



Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

D6.4 Compilation of internal deliverable outcomes ID6.13-6.18 (Learning activities and Units of Learning (main body of the report))

Due date of deliverable: 31-10-2009

Actual submission date: 13-01-2010

Start date of project: 01-12-2005

Duration: 4 years

University of Bolton

Version 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Project Deliverable Report

D6.4 – Compilation of internal deliverable outcomes ID6.7 – D6.18 (Learning activities and Units of Learning (main body of the report))

Work package	WP6 - Learning Activities & Units of Learning		
Task			
Date of delivery	Contractual: 31-10-2009	Actual: 13-01-2010	
Code name	D6.4	Version: 1.0	Draft <input type="checkbox"/> Final <input checked="" type="checkbox"/>
Type of milestone	Report		
Security (distribution level)	Public		
Contributors	David Griffiths, Davinia Hernandez, Mar Perez, Patricia Santos, Lau Llaubet, Javier Melero, Roy Cherian, Mar Perez, Michele Dicerto, Paul Sharples, Phillip Beauvoir, Scott Wilson, Kris Popat, Yongwu Miao		
Authors (Partner)	University of Bolton, OUNL, SU, FBM-UPF, GIUNTI		
Contact Person	David Griffiths (University of Bolton)		
WP/Task responsible	David Griffiths (University of Bolton)		
EC Project Officer	Mr. Martin Majék		
Abstract (for dissemination)	<p>This report describes the final phase of work on the TENCompetence Learning Design Toolkit, from July to December 2009. The work focused on the finalisation of the software produced, integration in the wider TENCompetence Personal Competence Manager, and deployment of the final system.</p> <p>In earlier releases the TENCompetence Learning Design Toolkit was only loosely coupled with the Personal Competence Manager. In this phase work was carried out to integrate the player applications used to deliver learning activities to learners into the Liferay system. Portlets were developed both for the player applications, and also to enable teachers to publish and populate Units of Learning for individuals or groups of learners.</p> <p>This work also involved further integration of the APIS runtime system for IMS QTI tests and questionnaires. This was complemented by the completion of finer grained integration between authoring of IMS LD and QTI, in the ReCourse Learning Design Editor.</p> <p>The Astro Learning Design Player was completed and delivered, meeting a need identified in earlier phases of the project for a replacement for the SLeD player.</p>		
Keywords List	IMS Learning Design Authoring, run-time IMS LD services, Astro, Units of Learning, ReCourse, Widget server		

TENCompetence Project Coordination at: Open University of the Netherlands

Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands

Tel: +31 45 5762605

Table of contents

1	Executive summary	4
2	Overview	6
2.1	The role of the IMS Learning Design and QTI in TENCompetence	6
2.2	The TENCompetence toolkit	7
2.3	WP6 tasks under DIP 4	11
3	Authoring	13
3.1	ReCourse enhancements	13
3.2	Localisation engine and translations	13
3.3	Templates	14
3.4	Documentation	17
4	Assessment	18
4.1	Included chapter: Support for interoperability and reusability of emerging forms of assessment: some issues on integrating IMS LD with IMS QTI	18
4.2	Self assessment tests for the Personal Competence Manager	36
4.3	Assessment templates	37
4.4	Running QTI assessments in the LD Toolkit	38
5	Learning design runtime	41
5.1	ASTRO Learning Design Player	41
5.2	Wookie	44
6	Integration	51
7	Evaluation	53
7.1	Evaluation of the LD Toolkit	53
7.2	Included paper: Evaluation of the Link Tool Portlet	53
8	Appendices	63
8.1	Appendix 1: List of templates	63
8.2	Appendix 2: ASTRO Learning Design Player quick start guide	71
8.3	Appendix 3: Wookie developers guide	79
8.4	Appendix 4: Evaluation of the LD Toolkit	98

1 Executive summary

This report describes the final phase of work on the TENCompetence Learning Design Toolkit, from July to December 2009. The work focused on the finalisation of the software produced, integration in the wider TENCompetence Personal Competence Manager, and deployment of the final system.

Throughout the project WP6 has sought to address the principal shortcomings of the infrastructure for using the IMS LD and QTI specifications to deliver learning and assessment activities. Having greatly enhanced the ease of authoring Units of Learning, and having resolved the lack of flexible runtime services for IMS Learning Design, work turned to the other major shortcoming identified by the project: the lack of an up-to-date, clear and attractive player with which to deliver learning activities. This work, the initial stages of which were described in D6.3, came to fruition in this period with the release of the Astro Learning Design player. This is the first general purpose open source learning design player to have been developed since SLeD, which was initially developed in 2004. Astro offers a full set of interface devices, and so much simpler user interfaces can be provided. It also offers a solid basis for other projects to build on, as was earlier the case with the SLeD player. This work is described in Chapter 5.

In earlier releases the TENCompetence Learning Design Toolkit had been only loosely coupled with the Personal Competence Manager. In this phase work was carried out to integrate the player applications used to deliver learning activities to learners into the Liferay system. To this end portlets were developed for the SLeD player, and also to enable teachers to publish and populate Units of Learning for individuals or groups of learners. This was challenging work, because the sophisticated functionality of the LD Toolkit needs to be integrated through web-services in order to keep the Personal Competence Manager as lean and efficient as possible. Integration work was tested by deployment of the systems in the Personal Competence Manager demonstrator in the final month of the project, now available as a publicly accessible system. In the final stage of work the new Astro player was also integrated with the QTI runtime system and the Personal Competence Manager. This work is described in Chapter 6.

This work also involved integration of the APIS runtime system for IMS QTI tests and questionnaires. The system was adapted for use with the Personal Competence Manager portlets, and portlets created to support user interaction. Further work on QTI enhanced support for the authoring of QTI, in the ReCourse Learning Design Editor. For a description of work in these areas please see Chapter 4.

The ReCourse Learning Design Editor has been extended and documented. Templates have been produced to assist authors, and integrated into the Editor. These cover a wide range of pedagogic and assessment techniques, including 360 degree and peer assessment examples. Many of the templates have been used in creating an eLearning demonstrator of the Personal Competence Manager, which will provide the Foundation with a means of demonstrating the relevance of TENCompetence to this area of application. This work is described in Chapter 3.

A significant achievement in this period has been the submission and acceptance of the Wookie Widget Server in the Apache Foundation Incubator, where a community of developers is now forming. It has also been recognised as a contribution to the emerging W3C Widget Specification, and has been integrated into a number of platforms beyond the project. For information about Wookie and the Apache Foundation please see Section 5.2.

Evaluation work has been carried out to gather the experiences of authors using the LD Toolkit to create and deliver learning and assessment activities (see Chapter 3). A pilot was also carried out to evaluate the use of the LD toolkit in the context of the Personal Competence Manager, and this is reported in a paper which is included in Chapter 6.

The system which has been developed in WP6 has made great strides in providing standards compliant, interoperable, sophisticated learning and assessment activities. It has been integrated into the Personal Competence Manager to provide a highly flexible environment for delivering competence development activities, also demonstrating how these activities could be integrated into other systems. The Open Source strategy followed by the project means that these achievements will be available to the wider community, through the TENCompetence Foundation, where they will become the reference point for future deployment and development.

2 Overview

2.1 The role of IMS Learning Design and QTI in TENCompetence

This overview of the role of IMS Learning Design and QTI in TENCompetence draws heavily on the discussion in D6.3. Some points are repeated here, rather than being referenced, so that this report may be read as a stand-alone document.

TENCompetence is committed to the use of interoperability specifications, and the systems it provides for the delivery of courses and assessments are based on IMS Learning Design (LD)¹ and IMS Question and Test Interoperability (QTI)². The tools provided by WP 6 support users in authoring, delivering and using learning activities and assessments which are compliant with these specifications.

A *learning activity* within the context of the TENCompetence domain model³ may be very *simple*, for example a web page of information together with an instruction to read the text and text some notes. Alternatively it may be *advanced*, that is to say, highly elaborated, for example with an extensive collection of study material with adaptive content and branching paths.

IMS LD enables the author to define a flow of learning activities called a *Unit of Learning* (UOL) which can be instantiated in *runs* for different cohorts of learners. The flow of activities is orchestrated by an *engine*, in our case (and almost universally for LD) Coppercore⁴. The user interacts with the UOL through a *player* application, which creates a user interface for the user, typically (though not necessarily) a browser. The IMS LD approach ensures that teachers can reuse the learning activities which they have carried out with one group of learners when they find need to repeat the course on a future occasion. It also offers a very high degree of flexibility in the pedagogic structures which can be implemented.

In the first phase of the project it was foreseen that both *simple* and *advanced* activities provided by TENCompetence would be delivered using IMS LD, including both simple learning activities. However, this approach was modified once the capabilities of the TENCompetence Personal Competence Manager (PCM) became clearer. The PCM provides the environment in which learning activities are embedded and contextualised in terms of competences, and it itself has the ability to deliver simple learning activities (see Vogten 2008⁵). The simplicity of these activities makes

¹ IMS. (2006). *IMS question & test interoperability specification v2.0/v2.1*. Retrieved 08/04, 2008, from <http://www.imsglobal.org/question/index.html>

² IMS. (2003). *IMS learning design specification*. Retrieved 08/04, 2008, from <http://www.imsglobal.org/learningdesign/index.html>

³ <http://hdl.handle.net/1820/649>

⁴ Vogten, H., Martens, H., Nadolski, R., Tattersall, C., van Rosmalen, P., & Koper, R. (2006). CopperCore Service Integration - Integrating IMS Learning Design and IMS Question and Test Interoperability. In *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies* (pp. 378-382). Kerkraade, the Netherlands: IEEE Computer Society.

⁵ Vogten, H., Koper, R., Martens, H., & Van Bruggen, J. (2008). Using the Personal Competence Manager as a complementary approach to IMS Learning Design authoring. *Interactive Learning Environments*, 16(1), 83-100.

it easier for teachers to create them as and when they are required, and the support for reuse can be provided by a repository of learning materials, ideally with support for searching distributed web based resources (as is the case with the LearnWeb2.0 application available integrated into the PCM).

Because of this it makes little sense to require authors to use an additional layer of IMS LD tooling when there is no need to support reusability of learning flows, and when simple learning activities can be achieved more simply with the native tools of the PCM. Moreover, the creation of learning activities with the PCM has an additional advantage of flexibility, as its simple activities can be created and edited at any time, while IMS LD requires a distinction between design time and run time.

As a result IMS LD is positioned within TENCompetence as an environment for the creation of sophisticated Units of Learning (UOLs), where the learning process to be orchestrated is sufficiently complex to require the use of the sophisticated capabilities of IMS LD in authoring, and where the time invested in preparation of UOLs makes reuse a significant issue.. Examples include UOLs which require multiple roles, adaptive provision of resources, and complex flows of activity. The TENCompetence LD Toolkit provides a highly flexible environment with which collaborative and adaptive Units of Learning (UOLs) can be authored and run. Using the roles, properties and conditions available in IMS LD pedagogic approaches can be implemented which are beyond the capabilities of Virtual Learning Environments.

2.2 The TENCompetence LD Toolkit

2.2.1 The significance of the toolkit

The success of this strategy beyond the life of the project, is closely tied to the adoption of IMS LD and QTI, and despite the valuable functionality which they offer, it must be accepted that adoption has been much slower than had been hoped for. The reasons why technologies thrive or wither are complex, and we have discussed them in Griffiths and Liber “The prospects for Learning Design”⁶. However, it is clear that one of the factors is that many of the applications available for making use of the specification have been no more than proof of concept tools, resulting from short research projects. The ambition of TENCompetence, and WP6 in particular, is to remedy this by creating a tool set which can be used as a delivery system in a real world context.

In general terms this involved a focus on improving the quality, usability and functionality, of the tool set, in terms of both functionality and usability. The TENCompetence strategy for supporting lifelong competence development is to provide an Open Source infrastructure which implements the TENCompetence domain model, and on which the wider community can build. In line with this the entire Learning Design Toolkit is composed of Open Source applications. **Where possible the project has extended and enhanced existing Open Source applications**, in particular the Coppercore Learning Design Engine, and the APIS

⁶ Griffiths, D. and O. Liber, Opportunities Achievements and Prospects for use of IMS LD, in *Handbook of Research on Learning Designs and Learning Objects*, L. Lockyer, et al., Editors. 2008, Information Science Reference.

QTI engine. Moreover, every effort has been made to encourage the engagement of the wider community with the new applications which have been produced by the project. In addition to making all code development available on SourceForge (in line with project policy), contacts with Apache led to the Wookie Widget server being invited to submit to the Apache Incubator. **Wookie has now been accepted into the incubator, and a community of developers has formed** around it who are mostly unconnected with the TENCompetence project⁷. translation component. As regards the ReCourse Learning Design Editor, this has been developed as a plug-in framework, although this is not visible to the user. Thus, for example, the QTI editor and the Visualiser (showing the structure of the UOL), and the Checker (parsing the UOL for correctness) are separate plug-ins which are integrated into the application. This makes it easy for other developers to extend the functionality, or indeed to replace or remove plug-ins. Moreover, requests were received from a number of developers asking if they could use the ReCourse libraries which handle the IMS Learning Design model in their own applications. This led to **refactoring of ReCourse so that a standalone Java library was used, which could be easily re-used in other applications**⁸. Similarly translation of ReCourse was facilitated not by creating a specific mechanism, but rather by using standard Eclipse language files and the **creation of a generic translation utility for Eclipse Rich Client Platform Applications**. This enables non experts to translate the application, and fills a gap in the wider Open Source infrastructure. This approach greatly enhances the prospects that **the resources put into the code produced by Work package 6 will provide value to the wider community following the close of the project**.

More specifically WP6 has **researched the challenges which were holding back adoption of IMS Learning Design and QTI as an effective means of delivering learning and assessment activities, and developed innovative solutions to them**. Specifically,

1. **Authoring of Units of Learning was too difficult for non experts to engage with**. The ReCourse Learning Design Editor, developed by TENCompetence, provides an environment which (a) **has a simple interface accessible to non-experts** who wish to author UOLs which include multiple roles and complex learning flows, and b) also **provides support for authors with the technical skills to edit the properties, conditions** which can be used to automate UOLs and create conditional structures. A **template mechanism and templates** have been included which simplify the task of authoring UOLs.
2. IMS LD was created by adapting OUNL EML. In this process **IMS stripped out all the assessment aspects of the specification** so as to avoid overlap with existing specifications, principally QTI. However, practical use of the two specifications together has been very demanding. The Learning Design Toolkit provides **improved integration of IMS LD and QTI in authoring and runtime**. The runtime environment developed in WP6 has also been **integrated into the PCM to deliver self-test competence assessments**.

⁷ See <http://incubator.apache.org/wookie/>. For the development mailing list and statistics see <http://markmail.org/search/?q=list:wookie>

⁸ Delivered in D6.3

3. The **poor integration of services in Units of Learning at runtime** (for example, chat, forums, wiki, etc.), required intensive development work on each target platform, creating a significant overhead to adoption. The **Wookie Widget server provides an effective solution to this problem**. It also has applications in other areas of eLearning and beyond, and has been recognised as a reference implementation of the W3C Widget Specification by W3C.
4. **The available players used out-of-date web technologies**. SLeD, the Learning Design player used by the project in pilots, is tied to a frame division of the screen and a tree metaphor for representing the Unit of Learning. It is also the result of extensive patching of what was originally a research tool rather than a delivery system. **An entirely new player called Astro has been developed as a replacement for SLeD, with a new approach to interfaces** for delivering Learning Design to learners.
5. As we point out in Griffiths and Liber (ibid.) the IMS LD specification was developed with the idea that the landscape of eLearning would consist of a relatively large number of competing Virtual Learning Environments (VLE). **It was (it seems mistakenly) assumed that the role of IMS LD would be to provide interoperability between VLEs**. At present, however, we find that the sector is dominated by one very large commercial vendor, and a few Open Source systems, led by Moodle. At the same time the VLE is being questioned, as a historically necessary but transient type of application, for example in the Personal Learning Environment proposed by Liber⁹, which shares much of its approach with the Personal Competence Manager. **Integration with the Personal Competence Manager demonstrates how learning activities can be disaggregated from the VLE, and be contextualised and delivered in other systems**.
6. **Administration of IMS LD systems required expert technical skills**, often using a command line interface. The toolkit has greatly simplified these tasks, providing direct mechanisms from the ReCourse authoring tool for **uploading to a repository or to a Coppercore runtime environment**. The LD Portlets enable teachers to easily **provide learning activities for groups of learners who are registered on the Personal Competence Manager**.

2.2.2 Toolkit components

The components of the TENCompetence Learning Design Toolkit are shown in the following table. New versions of all these components are provided in this deliverable, with the exception of the SLeD Learning Design Player. The links are to new or updated code which has been delivered in this reporting period.

The latest version of the source code is also available on the TENCompetence SourceForge site, and for easy download and installation the latest builds of the authoring and runtime components are provided at www.tencompetence.org/ldruntime and www.tencompetence.org/ldauthor

⁹ See Interactive Learning Environments Special Issue: Narrative and Interactive Learning Environments, volume 16 Issue 3, 2008. Liber, Oleg & Johnson, Mark (Eds.)

Table 1: Components of the LD Toolkit

Components of the TENCompetence Learning Design Toolkit		
Component	Description	URL
The ReCourse Learning Design Editor	An Eclipse Rich Client Platform application, with a plug-in architecture. Completely developed by TENCompetence.	http://hdl.handle.net/1820/2247
The QTI Editor	A plug-in for the ReCourse Learning Design Editor which enables authors to create QTI tests and assessments, and to integrate them with the learning flow of their Unit of Learning. Completely developed by TENCompetence	Delivered as a component of the ReCourse Editor above (http://hdl.handle.net/1820/2247)
Learning activity and assessment templates	A set of UOLs has been developed to demonstrate the Learning Design Toolkit, and these are made available to authors as templates from within the application, and as a collection of UOLs.	http://hdl.handle.net/1820/2283
Native Language Support tool	A tool developed to provide easy localisation of ReCourse. It is delivered as a separate utility for reuse with other Eclipse RCP applications.	http://hdl.handle.net/1820/2292
The NewAPIS Runtime Environment for QTI	This Runtime System interprets IMS QTI, renders it, manages user interaction and returns the results. An existing Open Source implementation which has been extensively adapted and improved to support IMS QTI 2.1, and to meet the requirements of the Personal Competence Manager	http://hdl.handle.net/1820/2279
The SLeD LD Player	The performance of the SLeD Learning Design Player has been enhanced, and its user interface enhanced. This existing Open Source player was used in the pilots of the TENCompetence system. However its old fashioned approach to web implementation mean that development work on recent releases of the toolkit has focused on the Astro player.	The last release of SLeD was made in D6.3.
The Astro LD Player	This Learning Design player was developed by the project to address the limitations of the SLeD Player. It provides a rich set of interface elements, and a well structured architecture and code.	http://hdl.handle.net/1820/2281
Learning Design Portlets for the Personal Competence Manager	These portlets enable the teacher to publish Units of Learning, and populate them with learners registered on the PCM. Learners can work with the activities in a Learning Design player portlet. Completely developed by TENCompetence	http://hdl.handle.net/1820/2291

2.3 WP6 tasks under DIP 4

The research and development challenges raised by the creation of an Open Source and standards compliant infrastructure for learning and assessment activities are substantial. Consequently it was important to identify those areas where project resources could most appropriately be applied. The project had to decide which existing Open Source applications could provide a sufficiently solid basis for development (e.g. Coppercore and APIS), and in which cases it was necessary to provide new solutions (e.g. Wookie and Astro). It was also necessary to prioritise the functionality which was required for lifelong competence development.

The strategy adopted was articulated in DIP 4, which guided development work. This restructured Work package 6 as a set of high priority tasks and task forces, situated within the existing tasks and objectives of the work package. This provided the framework for planning and carrying out the work reported in the present deliverable.

The **task forces** defined were as follows:

- Task Force 7 – QTI, responsible for integrating QTI tests into the Personal Competence Manager (situated in task 6.1)
- Task Force 13. Task Force 13 - ReCourse linking to Learning Objectives/Prerequisites and "Search Link to a UOL" Portlet (situated in task 6.3)
- Task Force 16 - Redesign SLED: Create a better, more stable, documented and modern version of SLED that is better integrated into the current set of TENC tools and services (situated in task 6.3)
- Task Force 18 - WYSIWYG XHTML editor for LD level B (situated in task 6.1).

The **objectives for the WP6 general tasks** were as follows:

Task 6.1: Authoring

- Extended support for creating Units of Learning.
- Authoring support for IMS-LD levels B and C.

Task 6.2: Assessment and QTI support

- Enhance the functionality of the QTI editor included in ReCourse, including
- Maintain and extend the APIS QTI engine, to support output of the QTI editor and ensure smooth integration
- Provide templates for the Assessment Specification.

Task 6.3: IMS-LD Runtime

- Develop a new player layer providing a user interface to the Coppercore Learning Design Engine.
- As required by pilots, develop additional widget services.

Task 6.4. Integration.

- An IMS LD administration portlet will be created.
- Integration of Learning Objectives with ReCourse.

- Provide a portlet and a link tool service that enables users to search for UOL IDs that can be included in any webpage (e.g. activities in the PDP) to link to that UOL.
- Integrate the Learning Design Toolkit runtime environment.

3 Authoring

3.1 ReCourse enhancements

A major release of ReCourse, version 2.0, was made in July 2009 (D6.3), and work in this period focused principally on localisation support, documentation, bugfixing, improved integration of the QTI authoring tool, and development and integration of templates.

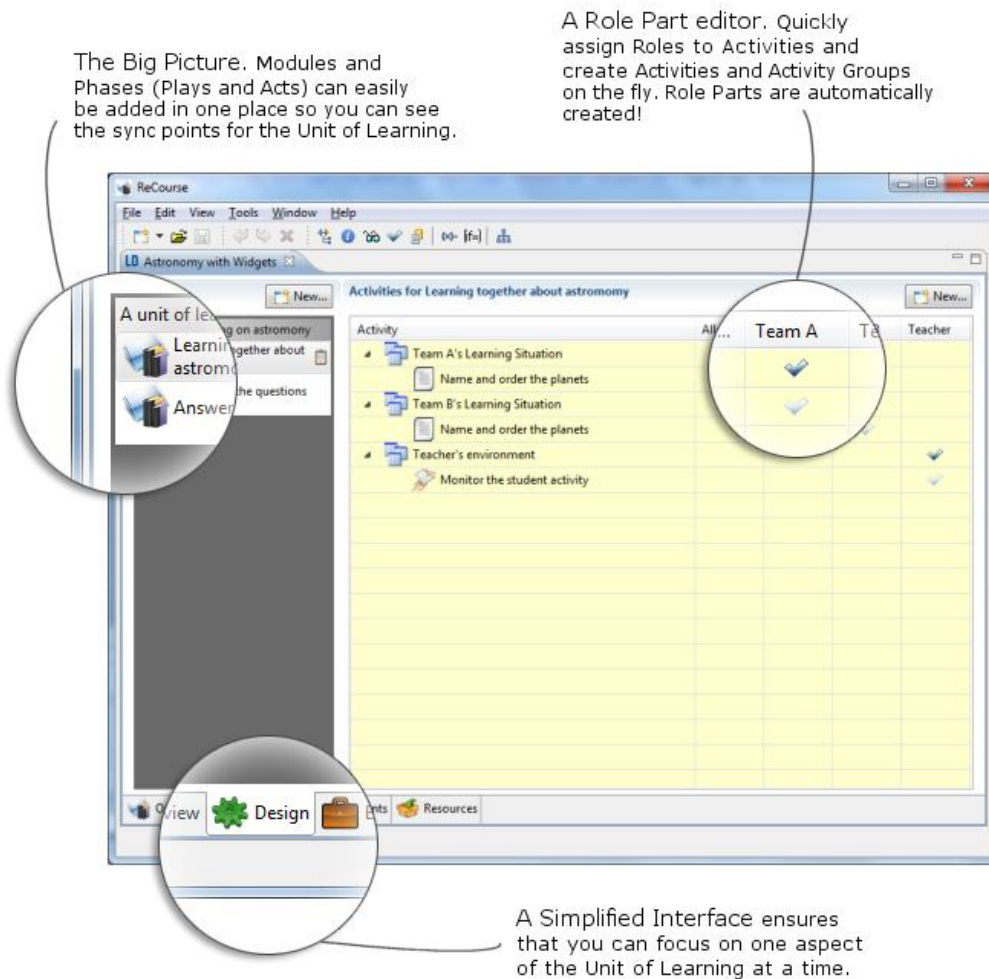


Figure 1: ReCourse Version 2.0

3.2 Localisation engine and translations

In order to maximise the potential adoption and maintenance of ReCourse it is important that it is available in as many languages as possible. However, localisation of an Eclipse Rich Client Platform (RCP) application such as ReCourse requires the translation of many strings in numerous *.properties. This creates a barrier to localisation, and a maintenance problem. In order to address this it was decided that rather than simply dedicating project resources to localisation of these strings, or creation of a *.properties editing capability in the application, it would be more effective, and little more time consuming, to resolve the underlying problem in the

Eclipse framework. Consequently the NLS (Native Language Support) Editor was developed in October 2009 and released at the beginning of November 2009 as a means to make the process of translating numerous strings in numerous **.properties* files easier for the translator¹⁰. This works with properties files produced by any Eclipse RCP application. The input is an English Language Pack archive file (*.pack) and the output is a Translated Language Pack archive file (*.translated).

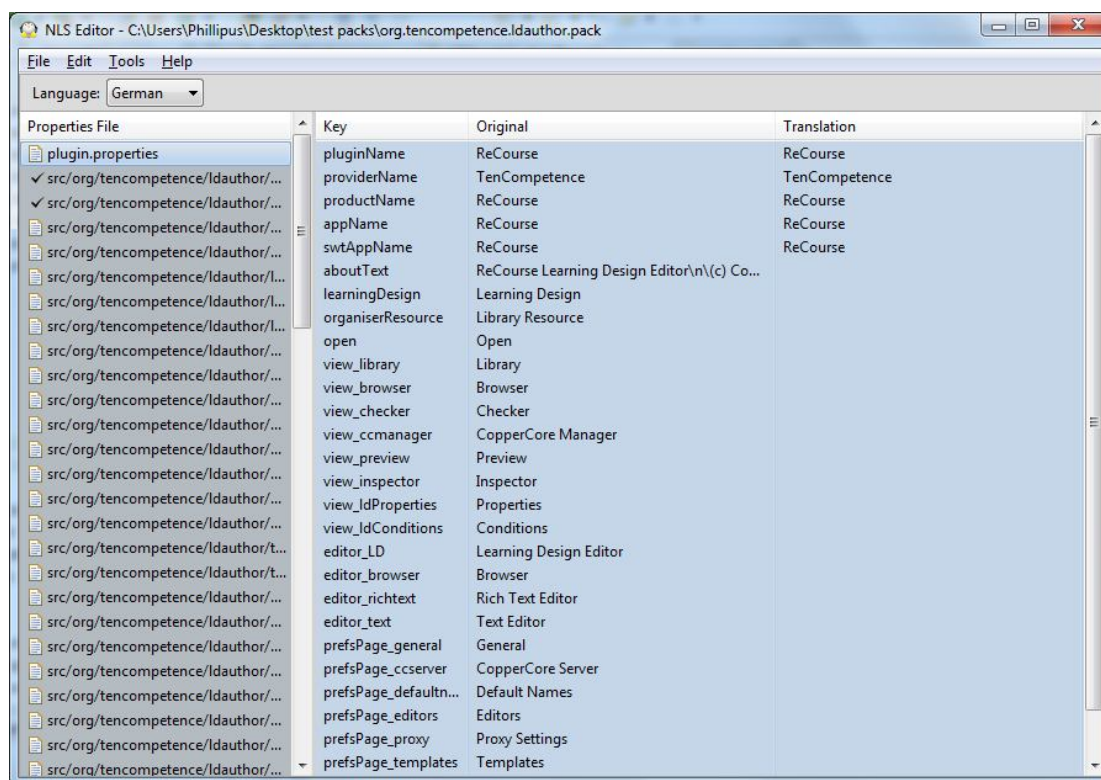


Figure 2: The ReCourse Native Language Support Editor

Each Language Pack archive file consists of a number of Properties files (shown in the left-hand table above). In turn, each Properties file consists of a number of key/value pairs. A translation can be left blank if required. This will mean that the original (English) value will be used. When all values have been translated for a Properties file, the icon in the left-hand table changes to a tick. Simple instructions for authors and developers are available at

http://www.tencompetence.org/ldauthor/nls_editor/

3.3 Templates

A mechanism has been built into the ReCourse Learning Design Editor which supports the creation of a template from an existing Unit of Learning. ReCourse ships with a set of Templates, but in this way authors can create their own to re-use and share. The process for creating a new template is as follows:

1. Create a new or open an existing Learning Design (UOL).

¹⁰ See http://www.tencompetence.org/ldauthor/nls_editor/

2. Edit the UOL so that it forms the basis for your template. Edit the description and the resources so that they may have a more general use.
3. Select "File->Save As Template..." from the main menu. A dialog opens (see figure 3).
Provide a name and description for the Template. Choose an existing category or create a new category from the "New..." button.
4. Press "Finish" to save the Template.

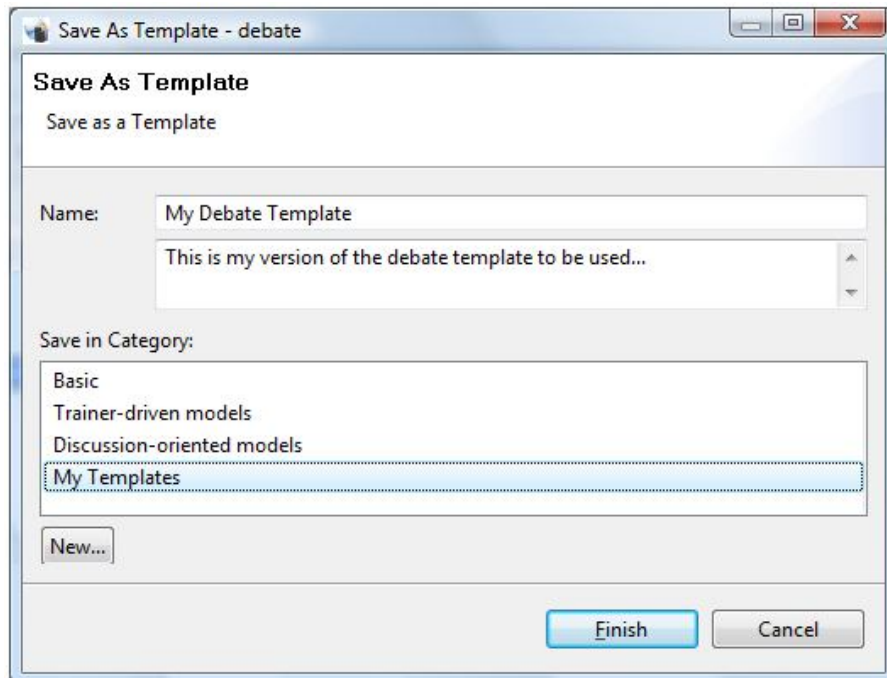


Figure 3: Saving a template in ReCourse

From now on, the new Template will appear in the "New Learning Design" wizard, as shown in Figure 4.

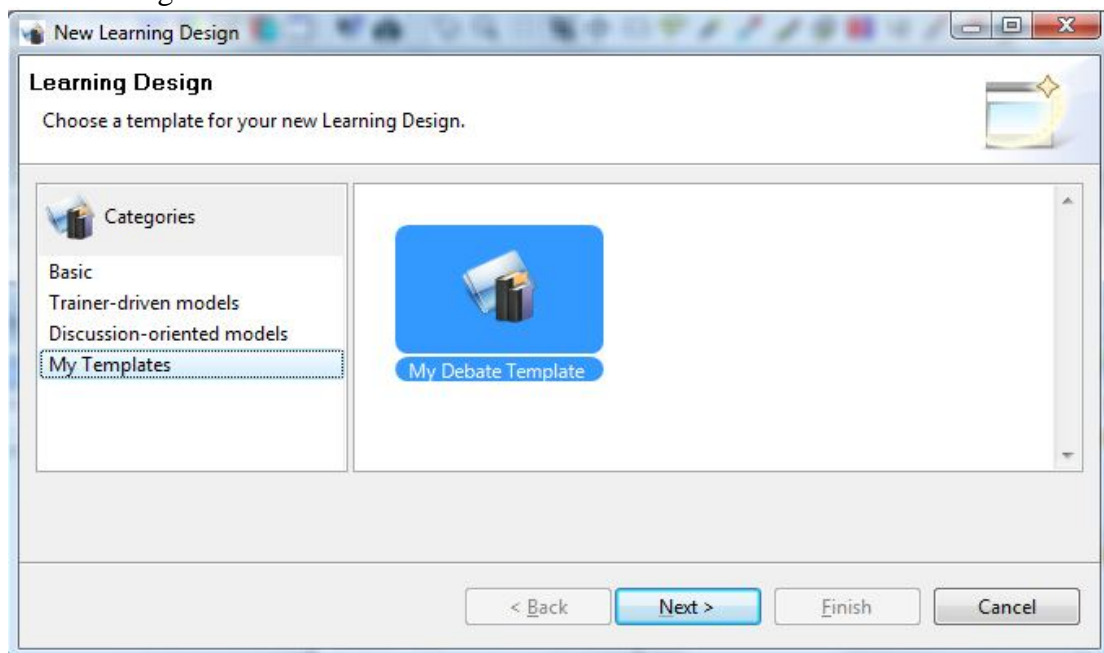


Figure 4: a user created template in ReCourse

Templates are managed from Preferences, enabling users to delete their Templates and Categories, edit the names and descriptions of User Templates, and change their Categories. However the built in templates cannot be changed.

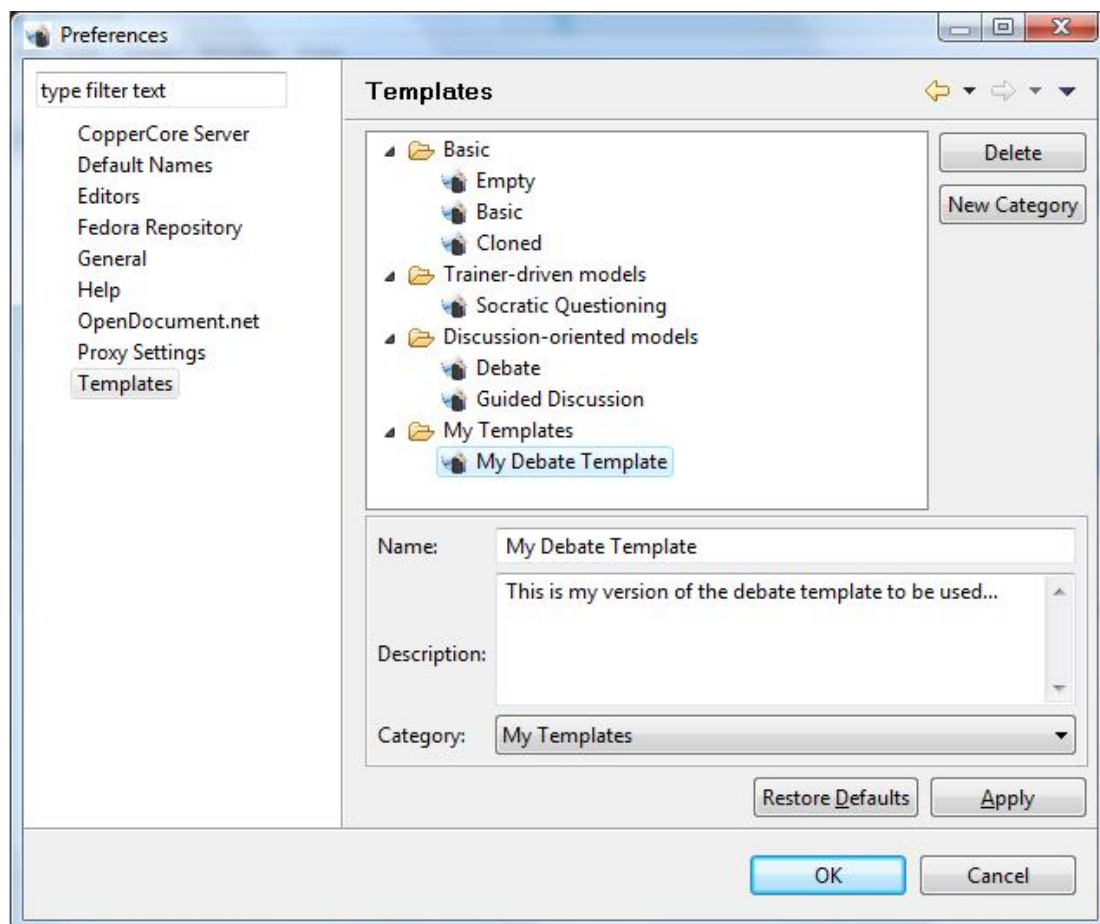


Figure 5: The template manager in ReCourse

Fourteen templates have been developed for ReCourse in this reporting period, selected from UOLs developed by the project, and these are provided in Appendix 2 to this report. These have been integrated into the application and are available to authors when they first open the application. The templates are also made available for separate download at <http://hdl.handle.net/1820/2283>.

The interoperability achieved by the ReCourse editor has meant that it has been possible to derive some of these from existing models. In particular the Prolix Project's Graphical Learning Modeller provides an editor which produces IMS LD. Unlike ReCourse, this uses its own file format for representing learning flows. This has advantages in representing certain pedagogic approaches, but it is more restricted in authoring and representing standard UOLs.

At a meeting of the JISC Learning Design Special Interest Group held at the Open University of the UK in June 2009 tests were successfully carried out to establish that UOLs created by the Graphical Learning Modeller could be exported, opened and edited in ReCourse. Contacts between the two projects have been maintained over this

reporting period, and Prolix have also shared with TENCompetence the pedagogic models which lie behind their specialised editors.

Because the graphical representations used by ReCourse are generated directly from the IMS LD Manifest, authors can use a range of compliant specialised authoring systems, export them, and then obtain a unified graphical representation and editing facilities in ReCourse. This cannot be achieved with other editors, because their graphical representations rely on custom file formats which are not interoperable. This capability of ReCourse greatly expands the number of templates available to authors, over and above those which are available natively in the application. Thus ReCourse has a role as the most effective way of viewing and editing the output of the template mechanisms of various specialised pedagogic modellers, such as the Graphic Learning Modeller.

3.4 Documentation

An online user guide was developed to assist new authors in working with the application. It is available at <http://www.tencompetence.org/ldauthor/user-guide.html>

4 Assessment

In this final reporting period work on assessment focused on three areas:

1. Development of portlets for authoring and delivering self-assessment tests
2. Adaptation of the new APIS runtime system so that it could provide assessment tests within the context of the Personal Competence Manager
3. Development of templates to demonstrate the use of the Learning Design Toolkit to deliver assessment activities.

The work described here has been informed by the research carried out within TENCompetence which is presented in the book chapter *Support for Interoperability and Reusability of Emerging Forms of Assessment: Some Issues on Integrating IMS LD with IMS QTI*, by Yongwu Miao et al. and published in the *Handbook of Research on E-Learning Standards and Interoperability*, 2009. The chapter is available from the TENCompetence Dspace repository at <http://hdl.handle.net/1820/2028>.

4.1 Included chapter: Support for Interoperability and Reusability of Emerging Forms of Assessment: Some Issues on Integrating IMS LD with IMS QTI

We now reproduce a pre-print version of this published chapter.

The numbering of the figures is conserved, but they are prefixed *Miao et al.*

The figures in the paper are not included in numbering the tables in the main body of this report.

Some Issues on Integrating IMS LD with IMS QTI

Yongwu Miao, Jo Boon, Marcel van der Klink, Peter Sloep, and Rob Koper

Centre for Learning Sciences and Technologies,

Open University of the Netherlands

Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands

ABSTRACT

In comparison with traditional assessment, emerging forms of assessment (e.g., self-/peer assessment and 360 degree assessment) involve multiple phases and multiple roles/persons, which are process-oriented assessment. IMS Question and Test Interoperability (QTI) is an open technical specification for task-oriented assessment, which has insufficient expressiveness to specify emerging forms of assessment. Meanwhile, existing software tools supporting emerging forms of assessment lack interoperability and reusability. In this chapter, we claim that a combined use of QTI and IMS Learning Design (LD) is able to support interoperability and reusability of emerging forms of assessment. In order to support this claim, we analyze the characteristics of four emerging forms of assessment from the perspective of process technologies and present the method to specify emerging forms of assessment using QTI and LD. Furthermore, we present the difficulties and problems that we encountered when modeling emerging forms of assessment and propose possible solutions to solve the problems.

INTRODUCTION

There is a marked tendency to place more and more emphasis on competences in education and, therefore, in assessment. Competence is defined as ‘effective overall performance within an occupation, which may range from the basic level of proficiency through to the highest level of excellence’ (Cheetham and Chivers 2005). A competence is the ability to handle a complex professional task by integrating the relevant cognitive, psychomotor and affective skills. Information gathering for the assessment of competences is increasingly based on qualitative, descriptive and narrative information, in addition to quantitative, numerical data. Such qualitative information cannot be judged against a simple, pre-defined standard (Vleuten and Schuwirth 2005). Some emerging forms of assessment have gained in acceptance and popularity in education. Examples of such forms of assessment are self- and peer assessment, accreditation of prior learning, and 360 degree assessment. These forms of assessments address complex traits of students and foster deep learning and the development of competences (Topping 1998; Boud, Cohen et al. 1999; Gipps 1999).

Assessment consists of making judgments (task aspect) and carrying out administrative activities (process aspect). In comparison with traditional assessment, both of these aspects of assessment are much more problematic in emerging forms of assessment. In particular, emerging forms of assessment usually involve multiple phases and multiple roles/persons. The difficulties and the potential for errors and omissions increase in a non-linear fashion as the number of candidates and assessors involved grows (Rosbottom 1994). As Bartram pointed out, 360 degree assessment by

its very nature is an administrative nightmare to manage. People involved in the process tend to be geographically dispersed but also need close supervision in order to ensure that the ratings are carried out to schedule and that sufficient raters are obtained for each focus of the assessment (Bartram 2005).

In order to make emerging forms of assessments work effectively and efficiently, many software tools have been developed and are increasingly being used. For example, MUCH (Rada, Acquah et al. 1993; Rushton, Ramsey et al. 1993), Peers (Ngu, Shepherd et al. 1995), Peer Grader (Gehringer 2001), SPARK (Freeman and McKenzie 2002), and ESpace (Volder et al. 2007) are multi-user tools that support self/ and peer-assessment. The eSPRAT system (Lockyer 2003; Davies and Archer 2005) and Appraisal360 (Appraisal360 home page) are example tools that support 360 degree assessment. In self- and peer assessment, with the help of software tools, the tutor, freed from administrative chores, is able to provide a useful, added-value service to students by acting as a problem solver. Student-assessors can concentrate on the clarity, correctness and completeness of each individual exercise without worrying about the relationship with other exercises (Rosbottom 1994). Similarly, for supporting 360 degree assessment, the software tools manage the workflow associated with the 360 degree assessment process, from initial setup and preparation of the people involved, through the management of the rating process (including delivery and scoring of questionnaires), to the production of reports and their delivery to feedback providers (Bartram 2005).

However, existing software tools supporting emerging forms of assessment are stand-alone and offer limited or no support for interoperability of systems and reusability of assessment resources. They each have their own data representation and their data are not interpretable and operable by other application tools. This prompts the question of whether existing e-learning technical specifications can be used to support emerging forms of assessment. The leading specification for the exchange and interoperability of assessments is IMS Question and Test Interoperability (IMS QTI, 2003). However, the QTI specification addresses the task aspect of assessment. Examples of specified assessment tasks are multiple choices, fillin- the-blank, and matching items. QTI provides no means to support the design and management of assessment processes. Specifically, it ignores who will be involved and what roles they will play at process level, what kinds of activities should be performed by whom and in which sequence, what assessment resources will be produced and used in which activities, and what dynamic changes may take place in the assessment process and under which conditions. In short, it provides insufficient support for the representation and execution of an assessment plan (Miao et. al. 2008). Therefore, QTI can not independently support emerging forms of assessment.

In QTI v2, the integration of QTI and IMS Learning Design (IMS LD, 2003) was specified. LD is an open e-learning technical specification that provides a pedagogy-neutral modeling language. It can be used to specify a teaching/learning process as a formal model, which can then be executed in a specification-complaint run-time environment (Koper and Olivier 2004). The integration between QTI and LD provides a possibility to technically model an aligned teaching, learning, and assessment process. However, only little reported work has been carried out on modeling emerging forms of assessment using LD and QTI. The objective of this chapter is to explore the possibility to support emerging forms of assessment by using existing e-learning technical specifications. Concretely speaking, we investigate the

expressiveness of LD and QTI in the representation of emerging forms of assessment by using a casebased analytical method. For each emerging form of assessment, we analyze the key features of the assessment from the perspective of process modeling and identify possible alternative scenarios in practice. We intend to share our experiences with readers in modeling emerging forms of assessment in LD and QTI. In addition, we will identify hurdles which may keep educators and assessment designers from using LD and QTI to specify their assessment. Finally, we propose possible solutions to overcome these difficulties.

BACKGROUND

Most open e-learning technical standards for course development and delivery (e.g., IEEE LOM, IMS CP, IMS SS, ADL SCORM) concern learning content (e.g., the description of content and the organizational structure of the content). Only IMS Simple Sequencing specification (IMS SS, 2001), which is also included in ADL SCORM, provides simple mechanisms to represent the sequence of content. In QTI v2, the integration between QTI and IMS SS has been specified as well. The effort has been made by the ASSIS project (ASSIS homepage) to integrate assessment into adaptive sequences of content. This approach enables a seamless integration between instruction and assessment and supports interoperability and reusability. However, such an approach assumes a learning model in which individual learners consume learning content with certain conditional control. It does not support the integration of learning activities with assessment activities. Instead, it just integrates learning materials with questions/questionnaires. The evaluation results of learners' answers are used to control the sequence of the presentation of the content, not the activity sequence. Therefore, it can not support emerging forms of assessment that involve multiple roles/users and complicated interactions among them.

In the development of e-learning technical standards, the release of LD signals an exciting paradigm shift from a content-centric approach to an activity-centric approach. LD provides a framework to express the pedagogical meaning of instructional content and in doing so reflects in a deeper and more creative way on how to design and structure activities (Koper and Olivier 2004). It can be used to specify a collaborative assessment process in which multiple people with diverse roles (e.g., designer, candidate, assessor, decision-maker, and other stakeholders) perform various activities (e.g., design assignment, create/collect evidence, evaluate evidence, and make decision) in sequence and/or in parallel coordinately at process level. However, LD can not explicitly support various types of assessment tasks. Assessment components within the Educational Modeling Language, the base of LD, were excluded when LD was adopted by IMS, because of the existence of QTI. QTI describes a data model for the representation of assessment item/test and the result report. It defines a set of interaction types which can be used to specify basic question types and complicated question types through combination. As mentioned before, QTI provides no support to model a multi-users/roles-involved and multi-phase assessment at process level. It is obvious that LD and QTI have their respective strengths and weaknesses when it comes to supporting emerging forms of assessment at process level and at task level. What is interesting is that their strengths and weaknesses are complementary. In the next section, we will examine whether an integration of LD and QTI can indeed support emerging forms of assessment.

A STANDARD-BASED APPROACH TO SUPPORT EMERGING FORMS OF ASSESSMENT

We present a standard-based approach to support four emerging forms of assessment: self assessment, peer assessment, accreditation of prior learning, and 360-degree assessment, because they are more and more important and popular in education. Table 1 shows the number of hits in the period 2004-2009 on the internet, using databases from Google Scholar and EBSCO.

Miao et al., Table 1. References to different assessment forms

	Google scholar (*)	EBSCO (**)
Self assessment	2490	65
Peer assessment	534	31
Accreditation of prior learning	18	0
360-degree assessment	27	2

(*) term in title, period 2004-2009,

(**) with option Keywords, January 2004 – June 2009P, peer reviewed articles, linked full texts.

Google Scholar is chosen because it reflects a more specialized collection, mainly consisting of empirical research articles. The ten articles found first in both databases are used as background for the descriptions of the assessment forms hereafter. Most of these articles offered some examples of assessment forms and these examples served as the input for the descriptions of the four assessment forms.

Characterizing Four Assessment Forms from Perspectives of Process Modeling

We first analyze the characteristics of the four forms of assessments from the perspective of process support technologies.

Key features of self assessment

Self assessment (SA) refers to a method where an individual assesses his or her own performance regarding a specific topic. The method is largely used both in work situations as in educational settings to initiate self reflection on issues related to performance. Also in many health related situations self assessment is a relevant method for self diagnosis. In work and educational situations the method is often combined with 360 degree assessment or with peer evaluation; in both cases the self assessment is a first step in the procedure, designed to make comparison with assessment of others and reflection on this comparison richer. The function of SA is evaluation or judgment of the worth of one's performance and the identification of one's strengths and weaknesses with the aim to improve one's learning/working/health outcomes. Table 2 lists the key features of SA from the perspectives of process modeling and alternative scenarios.

Miao et al. Table 2. Key features of Self Assessment

Roles	<ul style="list-style-type: none"> - the individual - the representative of the learning or working context; this might be <ul style="list-style-type: none"> - the teacher - the manager - the peers
Artifacts	<ul style="list-style-type: none"> - goals, criteria, procedures - scoring list or questionnaire on the relevant topics - evidence on performance using the scoring list or the questionnaire - the answers to the questionnaire or scores
Activities	<ul style="list-style-type: none"> - define the goal and the rules, criteria of the assessment. - score performance - report assessment result
Interaction	<ol style="list-style-type: none"> 1. Preparation: Representatives of the context together with the individual define the goal and the rules, criteria of the assessment. 2. Assessment: The individual scores him/herself on the relevant issues 3. Finalization: The individual communicate with others about the scores
Alternative scenarios	<ul style="list-style-type: none"> - Self assessment can be conducted by every individual without feedback to the organization. It is possible that a SA scenario has no the final phase. - Self assessment is often used as a first step in a process of 360 degree

Key features of peer assessment

Peer assessment (PA) can be characterized as the process in which students collaborate and evaluate their own performance as well as those of fellow-students (Sluijsmans et al., 2004; Gulikers, Sluijsmans, Baartman & Bartolo, 2009). Most implementations of peer assessment are not restricted to evaluating a peer's performance as such. In many educational contexts the basic idea is that it is essential that both actors, the candidate who undergoes the assessment (the assessed student) and the peers who conduct the peer assessment, should benefit from the peer assessment experience. Peer assessment is primarily used in professional and vocational education. In some professions there is growing interest in peer assessment (e.g. teaching profession) as a tool to enhance continuous professional development. Peer assessment is mostly used for formative assessment purposes, to provide students feedback on their performance that subsequently enables them to consider points of improvements for future learning experiences. Table 3 lists the components of a PA, the main procedures of PA, and alternative scenarios.

Miao et al. Table 3. Key features of peer assessment

Roles	<ul style="list-style-type: none"> - teacher - candidate - assessor
Artifacts	<ul style="list-style-type: none"> - instruction - standards and criteria - evidence - assessment form - feedback - improvement
Activities	<ul style="list-style-type: none"> - inform students - group students - create evidence - assess evidence - evaluate feedback - compose points of improvement
Interaction	<ol style="list-style-type: none"> 1. Preparation Teacher informs students about goals, procedures, timelines etceteras; Teacher groups students in pairs, trios or larger groups. 2. Creating evidence Candidate uses instruction, the standards and criteria to create the evidence. 3. Assessing evidence Peers use the instruction, the standards and criteria to evaluate candidate's performance; Peers fulfill assessment form and write feedback. 4. Reaction Candidate evaluates the feedback and composes points of improvement;
Alternative scenarios	<ul style="list-style-type: none"> - Peer assessment is often reciprocal, i.e. after the first round, roles shift, and that the candidates subsequently become peers and vice versa. - There is not always written evidence. In some cases, peers observe the behavior of the candidate, which then is the evidence to be judged (for example student teachers who assess each other during internships in schools) - In many cases peers are required to reflect on their role as peer assessor. - Sometimes the candidate informs peers on the quality of feedback received

Key features of accreditation of prior learning

Accreditation of prior learning (APL) supports lifelong learning by assessing and recognizing someone's competences obtained informally through (paid and unpaid) work experiences (Joosten-ten Brinke, 2008). APL is most often offered by educational providers who promote APL in order to attract non-traditional student groups. APL is primarily used in education as a means to determine the content and size of student's study program prior to study entrance. It is a sound instrument only for employees who possess sufficient work experience in the domain they want to be educated for. APL is mainly used for summative assessments. Outcomes of the APL procedures are utilized by examination boards to determine what needs to be learned by prospective students in order to receive a particular certificate or diploma. In Table

4, we present the key features of APL from the perspectives of process modeling and alternative scenarios.

Miao et al. Table 4. Key features of procedures for accreditation of prior learning (APL)

Roles	<ul style="list-style-type: none"> - mentor - assessor - employee's (prior) employer - employee (hereafter candidate) - examination board
Artifacts	<ul style="list-style-type: none"> - description of set of competencies, including standards and requirements for portfolio - evidence and portfolio - form to check candidate's portfolio - rubrics and scoring forms for assessors - APL certificate - form to notify candidate on study program reduction - form for candidates to appeal against the outcome of their APL procedure
Activities	<ul style="list-style-type: none"> - discusses - select the competences - collect evidence and store in a portfolio - check portfolio - assessed portfolio using rubrics and scoring forms. - write report (APL certificate) - decide to what extent it is allowed to reduce the candidate's study program.
Interaction	<p>1. Candidate-profiling Candidate discusses with mentor the possibilities for APL; Candidate receives description of set of competencies, including standards and requirements for portfolio.</p> <p>2. Evidence gathering Candidates collect and classify evidence about their previous experience; Mentor checks the content of candidate's portfolio.</p> <p>3. Assessment Assessors review the quality of a candidate's evidence using assessment standards and rubrics; Candidate receives a report that describes to what extent the candidate master the competences that are included in the competence profile.</p> <p>2. Recognition Assessors compose APL certificate and send to candidate; Candidate send APL certificate to examination board; Examination board notify candidate about decision on study program reduction.</p>
Alternative scenarios	<ul style="list-style-type: none"> - Candidates assess their own prior experience in light of the standard and include the outcomes of this self-assessment in the portfolio; - Besides portfolio assessment one or more additional assessment activities usually will take place, such as a criterion-based interview, demonstration, knowledge test.

Key features of 360 degree assessment

360 degree assessment is also known as multi-source performance assessment or 360 degree feedback. The method refers to the process by which performance appraisals are collected from different sources, such as supervisors, peers, subordinates and sometimes also customers - rather than from a single source. This should provide the feedback recipient with a unique combination of information which is not otherwise available. It is assumed that the feedback givers chosen are in the best position to observe and evaluate certain types of behaviors. The method can be used for assessing performance and designing professionalization or development paths, sometimes the method is used to analyze interpersonal behavior (Whitehouse et al., 2007) or for training evaluation (Jellema, Visscher and Scheerens, 2006). It is used sometimes as a decision making tool (for example on career advancements or salary increases). 360 degree assessment is usually used at workplaces, both private and public. It can also be used in a class situation for educational purposes, but this is less likely. Table 5 shows the key features of 360 degree assessment.

Miao et al. Table 5. Key features of 360 degree Assessment

Roles	<ul style="list-style-type: none"> - feedback receiver (or target employee) - responsible for process (RFP), can be a HRM representative - feedback giver: <ul style="list-style-type: none"> - supervisor - peers/co-worker - subordinate
Artifacts	<ul style="list-style-type: none"> - form with closed and open questions on issues and criteria to be used as a questionnaire or a guide for an interview - mission statement of organization with competency map - appraisal and feedback - summary and priorities
Activities	<ul style="list-style-type: none"> - define assessment goals - instruct - formulate appraisal - structure feedback - communicate feedback
Interaction	<ol style="list-style-type: none"> 1. Preparation <ul style="list-style-type: none"> - HRM representative define assessment goals; - HRM representative instructs all participants on procedure, roles, goals and criteria. 2. Assessment <ul style="list-style-type: none"> - Downward appraisal from supervisor - Lateral appraisal from peers/co-workers - Upward appraisal from subordinates - Inward appraisal from target employee 3. Finalization <ul style="list-style-type: none"> - HRM representative summarizes feedback - HRM representative formulates next steps trajectory

Roles	<ul style="list-style-type: none"> - feedback receiver (or target employee) - responsible for process (RFP), can be a HRM representative - feedback giver: <ul style="list-style-type: none"> - supervisor - peers/co-worker - subordinate
Alternative scenarios	<ul style="list-style-type: none"> - The target employee formulates improvement goals at the beginning of the process and the different other roles react on these - The input from each appraisal is discussed consequently with the target employee - The target employee gives feedback on improvement goals to the superior, peer or subordinate - The self assessment is not always part of the procedure. Some authors argue that a previous self assessment optimizes the process (Garbett et al., 2007) - More than one employee from each role-group is appointed (more than 1 supervisor, peer, subordinate) - Feedback can be given during a group session; this could reinforce the effects of reflection (van der Heijden and Nijhof, 2004). - Feedback can be given anonymous or anonymously - A group of employees instead of a target employee can be the feedback receiver - A training is given to participants if necessary - Some companies collect feedback from the customer

Modeling Emerging forms of Assessment Using LD and QTI

QTI v2 specified integration of LD with QTI by coupling an LD property to a QTI outcome variable. The original motivation for integrating LD and QTI stems from use cases involving formative assessment and summative assessment using assessment items with traditional question types. Here we try to extend the application areas of the integration of LD and QTI and to improve the benefit of their combined use. As a consequence, the emerging forms of assessment can be modeled as a unit of assessment, a processoriented assessment model represented in the form of a specific unit of learning. Thus, such a unit of assessment can be executed in an LD and QTI compliant run-time environment. Furthermore, a unit of assessment can be instantiated as a complete model many times and can be customized or partially reused by different groups/organizations.

When analyzing the emerging forms of assessment, we have created a table for each form of assessment in the last sub-section. There are five rows in each table: roles, artifacts, activities, interaction, and alternative scenarios. The first three rows are components of a process. The interaction describes how participants with diverse roles perform activities in sequence and/or in parallel and how artifacts are used, produced, and transferred in/between activities. Alternative scenarios describe some variations in assessment practices. In this sub-section, we present how to model them through a combined use of LD and QTI.

Modeling multiple roles

As we have seen in each table, multiple roles are involved in each form of assessment. When modelling an emerging form of assessment, it is required to explicitly define multiple roles. The QTI specification is concerned with individual learners. Although QTI does not prohibit use in contexts involving other actors (e.g., instructors, supervisors, and peers), it does not explicitly support defining other roles or sequencing behaviors that result from participation of other actors. However, LD can support a multi-role/user teaching-learning process. In LD, two primary roles (learner and staff) are pre-defined. Each role can have sub-roles defined by designers to fit the context of the learning design. A role is bound with certain activities as role-parts. At run-time a person with a certain role will have privileges and responsibilities which allow him or her to perform the activities and to access certain learning resources according to the definition of the learning design. With LD, multiple roles as listed in the four tables can be modeled. The hierarchical structure of roles (e.g., in 360 degree assessment the role of feedback giver has three subroles: supervisor, peers/co-worker, and subordinate) can be modeled as well. Note that in LD each role can be played by multiple users at run-time. Thus, it can be modeled that more than one employee from each sub-role of feedback giver can be appointed in 360 degree assessment.

Modeling artifacts

In each emerging form of assessment various types of artifacts are created and/or used in activities. Some are represented in the form of questions (e.g., some assignment forms for creating evidence and some assessment forms with rubrics) and some are normal documents for different purposes (e.g., assessment goal and feedback). Usually, an artifact in the form of question/questionnaire can be modeled using QTI, which can represent many types of questions such as multiple-choice/response, Likert-scale, openquestion, fill-in-blank, hot-spot, matching, ordering, association, slider, drag&drop, and upload-file. QTI also provides sufficient flexibility to grow into the advanced constructed-response items and interactive tasks we envisage as the future of assessment (Almond, Steinberg et al. 2001). Furthermore, it provides mechanisms to design structured assessment and control branches and calculate weighted scores. That is, all standard questions and structured tests/exams that form the core of current practice can be supported by using QTI. In addition, LD can be used to represent non-question artifacts. Although LD has no concept of “artifact” in the specification, it enables to define a property with a data type, such as string, text, Boolean, integer, real, url, time, duration, and file. A kind of artifact can be modeled as a property using an appropriate data type. For example, an assessment goal or a feedback item can be defined as a property with the string or text type. A structured document can be modeled as a file-type property. Note that reusable documents can be put on the web and can be accessed by many assessment processes through using URLs of the web pages.

Modeling activities

In each emerging form of assessment, various activities are performed by diverse roles. LD provides constructor (i.e. activity and environment) to define an activity with some attributes (i.e., title, description, and completion). Most activities listed in the tables can be easily modeled in LD through specifying the values of attributes. For modeling some assessment activities, the question/questionnaire should be modeled as a QTI document as described above, which has to be referred to by an information

item within the activity or in the associated environment. It is important to note that a corresponding LD property should be defined in such a way that its identifier is a combination of the identifier of the QTI document and the identifier of the corresponding outcome variable, such as a score. When a candidate/assessor accesses the activity or the environment at run-time, the question/questionnaire will be presented to the candidate/assessor by the QTI engine. After the candidate/assessor submits the answer(s), the QTI engine will evaluate the response and transfer the result to the LD engine. Then LD engine can then adapt the teaching/learning process to the assessment result. For supporting some online activities, such as interview, monitoring, and group meetings, additional services are needed. Fortunately, LD provides some built-in services such as conference and monitor, which can be used to support online communications and monitoring works of participants with a given role.

Modeling interaction

As illustrated in the tables, emerging forms of assessment are phase-based processes, in which multiple participants with diverse roles perform various activities in sequence and/or in parallel and artifacts are transferred from one activity/role to another.

QTI allows candidates to answer questions in a pre-defined sequence or in any order to finish an assessment test. However, such control of the sequence of the tasks is restricted within an individual assessment test. LD can support the modeling of a learning flow with complicated process controls. Activities can be arranged as a sequence or a selection structure. A set of role-parts can be performed in parallel within an act, and acts within a play will be carried out in sequence. Multiple plays can be executed as concurrent threads. The termination of one activity may trigger the start of another activity. In addition, conditions and notifications provide more powerful mechanisms to control the process. The support provided at LD levels B and C makes it possible to trigger the start and termination of activities in a data-driven manner as well.

Some artifacts such as evidence and feedback are intermediate products, which are transferred from one activity/role to another. Some are pre-defined and assessable in the assessment process. QTI provides mechanisms for declaring outcomes. The outcome of an item, a section or a test can be processed as the output of an assessment. QTIv2 specifies how an outcome variable of QTI can be coupled to a LD property. With the help of this mechanism, an item response and an assessment score can be transferred to relevant participants. That is, the data produced by a participant (e.g., a candidate) can be presented to another one (e.g., an assessor). Additionally, scores given by all assessors can be processed according predefined calculation rules as a final result. This result can be transferred to a candidate or even can be used to control the branching. Furthermore, LD provides rich mechanisms to produce and transfer artifacts that are modeled as properties. For example, set-property, change-property, and view-property are basic mechanisms to create, modify, and retrieve artifacts. The local property and global property allow one to transfer artifacts within a learning design and across learning designs. The monitor service can support to view the artifacts produced by other roles.

In summary, both LD and QTI have certain strengths and weaknesses in their support of emerging forms of assessment. They cannot model all features of emerging forms

of assessment independently. However, they complement each other on task and process aspects. Thus a combined use of LD and QTI can model most of the features of emerging forms of assessment listed in the tables. In the next sub-section, we will use this standard-based method to model an example of an emerging form of assessment.

An Example

In this sub-section we describe a 360 degree assessment scenario. Then we model it with LD and QTI and present how to execute it.

Description of a 360 degree assessment scenario

Professor Hicks works at department C of a university and is responsible for the coordination of one of the sections of this department, focusing on the theme of consumer education. He develops research proposals and acquires research funds, supervises young researchers and has contacts with paying clients outside the university who want to have his advice on consumer education. He has three senior researchers who support him in his job.

In the department where he works a competence map is developed that describes all the competences relevant for different staff members in different jobs. In the beginning of the year, the management team decides that a new round of 360 degree assessments will be organized. The staff member who is responsible for the coordination of sections sends mister Hicks a mail explaining the procedure, and setting a time frame for about when he will have a talk with his manager, in his case the director of the department.

First, professor Hicks uses the competence map to perform a self assessment. Using the map he rates his score on the relevant competences and decides on which topics he would like to have more formal and informal training in the coming year. Second, he invites a coordinator of another section, one of the young researchers he is supervising, as well as one of the clients he worked for during the last months. He asks all three feedback givers to fill in a short questionnaire with questions on his commitment, the quality of his output, the degree to which he keeps his appointments and the quality of his functioning as a team member. The questionnaire leaves room for other remarks on his performance. Three feedback givers send their reactions to the director of the department and send a copy to professor Hicks himself.

At the agreed date, the director of the department receives the self assessment and the information of the three feedback givers and the report of the 360 degree assessment of last year. He uses all this information to have a discussion with professor Hicks about his performance. In the self assessment Hicks indicates some competences on the management level where he wants to have some training, especially on the field of supervision and time management. It turns out that his colleague coordinator is very positive on all points and only mentions that sticking to appointments is sometimes a problem; professor Hicks often comes late in meetings and has to leave early. The young researcher is also very positive but mentions that she has to wait sometimes for weeks before receiving feedback on research proposals. The client is very satisfied on all the points and mentions that for the next contract he wants professor Hicks to advise him on a specific new topic. During the discussion with the director of the department appointments are made about training in time management, delegation of tasks and

setting of priorities. The appointments are formalized in a short report and stored in the personnel portfolio of professor Hiks.

Modeling the 360 degree assessment scenario

We can develop a descriptive model that formally specifies the scenario with LD and QTI. A descriptive model abstractly describes how a process is performed in a particular environment in an inductive manner. In the model, five roles are defined: feedback receiver, manager, and three feedback givers including colleague, subordinate, and client. The competence map is modeled as a QTI test document including a list of Likert-scale questions. Three short questionnaires for feedback givers are modeled as QTI test documents as well. The reports of the 360 degree assessment of last year and this year are modeled as file-type properties. Five activities are defined: one self-assessment, three assessment activities of feedback givers, one discussion. The whole process consists of three phases: self assessment, assessment of feedback givers, and discussion and decision. Self assessment result and all feedbacks created in the first two phases will be used in the discussion. A short report will be produced in the discussion.

Execution of the model and reuse of the model

The model can be published in a LD and QTI compliant run-time environment. If the assessment would be conducted in the computer-supported environment, the process will be carried out as below.

The staff member who is responsible for the coordination of sections, instantiates the model by creating a new run of the model. S/he has to prepare settings for this run through assigning the role of feedback receiver to Hiks and assigning the role of manager to the director of the department. The staff member will arrange a conference service if the discussion is an online activity. Otherwise, a meeting room should be arranged with a scheduled duration for the discussion. After that, the staff will inform all about the start of the assessment. Professor Hiks can access the first activity in which the instruction about how to carry out the assessment and the competence map are available. The expected output of this activity is the self assessment result. Then, he invites three participants by assigning the role of colleague to the coordinator, the role of subordinate to the young researcher, and the role of client to the person for whom professor Hiks has worked during the last months. The invited feedback givers will be informed and can find an assessment activity in their to-do list. After accessing the activity, s/he can read the instruction and the short questionnaire. After having answered the questionnaire, s/he can simply submit it. All assessment results and the report of 360 degree assessment of the last year can be accessed in the discussion activity. In the time scheduled, Hiks and the director of the department can access the activity work space and discuss results either using the online service or face-to-face. The director can write a short report in the activity work space and the report will be send to professor Hiks. This then terminates the execution of the assessment.

It is important to note that this model can be reused for assessing other colleagues of the department. For this purpose, the staff member only needs to create other runs and to assign the role of feedback receiver to other colleagues. The model can also be reused for assessing the performance of professor Hiks in the next year. Finally, it can be customized by other departments through modifying the competence map and questionnaires.

FUTURE RESEARCH DIRECTIONS

When modeling emerging forms of assessment, we encountered some difficulties and problems. Firstly, it is difficult to perform statistical analyses (By statistical analyses we do not refer to the usual analysis of assessment results, but rather data analyses that lead to an adaptation of the assessment process itself), if the number of role members is not fixed in an assessment process. Even if the number of candidates is predictable, the degree of complexity of the model will increase as the number increases. For example, if the number of peers is unpredictable, the score given by each peer can only be modeled as a personal property. However, LD provides no means to express the calculation of the mean of the scores given by all peers.

Secondly, the adaptation of an assessment is currently restricted within the definition of the assessment and the assessment can be adapted only to candidates' responses to the questions. It is difficult to adapt assessment to the learners' characteristics and environmental information. For example, the competence map cannot be adapted to the position/function of the feedback receiver. Thirdly, assignments and/or assessment forms, sometimes, have to be developed by the participants at run-time, not by the designer at design-time. It is difficult to include new assessment after a UoL has been published. For example, in APL it is unpredictable what additional questions are required to answer. The assessor may need to create a questionnaire for collecting additional evidence at run-time. Fourthly, it is difficult to integrate assessment-specific services in LD. For example, in APL additional assessment activities may be needed in which assessment-specific services such as certain simulators and conceptmapping tools are needed. In the near future, research should target solving the problems just identified, if we want genuinely to support emerging forms of assessment in an interoperable and reusable manner. Firstly, LD would have to be able better to deal with personal properties (e.g., the sum of scores given by multiple peers when the score is modeled as a personal property); this can be done by extending the specification of the expression element. Secondly, the concept of 'income variable' should be introduced in QTI, so that the information can be transferred from teaching-learning activities to assessment. The adaptation can be defined in such a way that it adapts assessment to the value of income variable. Thirdly, QTI editor had better be specified as a built-in service in LD, so that LD can handle the QTI documents created at the run-time. Fourthly, a more generic solution (like BP4WS in business process management) should be developed to integrate third-part services in LD, so that the external services can be specified in the design-time, can be configured at instantiation-time, and then can be invoked at the run-time easily.

Finally, the standard-based approach for modeling emerging forms of assessment described in this chapter suits technical developers only, who have a sound knowledge of process modeling and technical specifications. As pointed in (Miao and Koper 2007), it is very difficult if not impossible for practitioners to model a complicated teaching, learning, and assessment process with LD and QTI. In order to support 'ordinary' teachers and assessment designers to specify and customize an assessment plan, a high-level assessment modeling language is needed; this we are currently working on (Miao et al. 2008 and Miao et. al. in press). For the sake of interoperability and reusability, an assessment plan represented in such a high-level modeling language will be transformed into an executable model represented in LD

and QTI. Thus the assessment process can be supported by using existing LD and QTI compliant run-time environment.

CONCLUSION

Emerging forms of assessment become increasingly important. The importance of four emerging forms of assessment is underpinned by presenting some findings on how broadly these forms are represented in contemporary academic writings.

Through an analysis of key features of four emerging forms of assessment from the perspective of process technologies, we found that all these forms of assessment 1) involve multiple roles/participants; 2) deal with various artifacts; 3) consist of various activities; and 4) include a complicated control-flow and dataflow. Although many software tools have been developed to support emerging forms of assessments, these software tools are stand-alone and lack interoperability and reusability. QTI, the leading specification for the exchange and interoperability of assessments, supports task-oriented assessment, but cannot support process-oriented assessment. LD, a process-oriented modeling language, can be used to model multi-role/user and multi-phase processes, but lacks facilities to model various assessment tasks. That is, neither of them can fully support emerging forms of assessment.

In this chapter we developed and presented an approach to support interoperability and reusability of emerging forms of assessment. The approach is based on the existing open e-learning standards LD and QTI. Through a combined use of LD and QTI, emerging forms of assessment can be modeled as units of assessment, which then can be executed in any LD and QTI compliant run-time environment. That is, an emerging form of assessment represented as an executable model can be reused and customized by other groups/organizations. Meanwhile, the components of a model can be reused as well. Because the model is represented in LD and QTI, all standard-compliant tools (irrespective of the authoring tool, repository, simulator, or engine) can interoperate on the assessment model.

We also indicated some difficulties we met when modeling emerging forms of assessment with LD and QTI. We proposed solutions to overcome them. As part of that, we are working on a high-level, assessment process modelling language. It is designed for practitioners to allow them to specify or customize emerging forms of assessment. Using this language, the emerging form of assessment can be specified as a high-level assessment process model, which can be automatically transformed into an executable model represented in LD and QTI. Once this goal is achieved, practitioners will be able to reap the benefits from using technical standards without the need to handle technical complexity.

REFERENCES

- Appraisal360, home page, Retrieved July 12, 2009, from <http://www.appraisal360.co.uk/home>
- ASSIS, home page, Retrieved July 12, 2009, from <http://www.hull.ac.uk/esig/assis.html>
- Almond, R. G., L. Steinberg, et al. (2001). A sample assessment using the four process framework. CSE Report 543. Center for study of evaluation. . Los Angeles, University of California.
- Bartram, D. (2005). Testing on the Internet: Issues, Challenges and Opportunities in the Field of Occupational Assessment. Computer-Based Testing and the Internet: Issues and Advances. D. Bartram and R. Hambleton (Eds), Wiley.
- Boud, D., Cohen, R. et al. (1999). Peer Learning and Assessment. *Assessment and Evaluation in Higher Education* 24(4): 413-426.
- Cheetham, G. and G. Chivers (2005). Professions, Competence and Informal Learning. p. 54, Cheltenham, Edward Elgar Publishing.
- Davies, H. and J. Archer (2005). Multi-source feedback: development and practical aspects. *Clinical Teacher* 2(2): 77-81.
- Freeman, M. and J. McKenzie (2002). Implementing and evaluating SPARK, a confidential web-based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects. *British Journal of Educational Technology* 33(5): 553-572.
- Garbett, R., Hardy, S., Manley, K., Titchen, A., McCormack, B. (2007). Developing a qualitative approach to 360-degree feedback to aid understanding and development of clinical expertise. *Journal of nursing management*, 15, 342-347.
- Gehringer, E. F. (2001). Electronic peer review and peer grading in computer-science courses. *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Charlotte, North Carolina.
- Gipps, C. (1999). Socio-cultural perspective on assessment. *Review of Research in Education* 24: 355- 392.
- Gulikers, J., Sluijsmans, D., Baartman, L. and Bartolo, P. (2009). The power of assessment in teacher education. In A. Swennen and M. van der Klink (eds.) *Becoming a teacher educator. Theory and practice for teacher educators*, 173-188. Dordrecht: Springer
- IMS LD. (2003). IMS Learning Design Specification. Retrieved July 12, 2009, from <http://www.imsglobal.org/learningdesign/index.cfm>.
- IMS QTI. (2003). IMS Question and Test Interoperability Specification. Retrieved July 12, 2009, from <http://www.imsglobal.org/question/index.html>.
- IMS SS. (2003). IMS Simple Sequencing Specification. Retrieved July 12, 2009, from <http://www.imsglobal.org/simplesequencing/index.html>
- Jellema, F., Visscher, A., Scheerens, J. (2006). Measuring change in work behavior by means of multisource feedback. *International Journal of Training and Development*, 10(2), 121-139.

Joosten-ten Brinke, D. (2008). Assessment of prior learning. Heerlen, The Netherlands: Open Universiteit Nederland

Koper, R. and B. Olivier (2004). Representing the Learning Design of Units of Learning. *Journal of Educational Technology & Society* 7(3): 97-111.

Lockyer, J. (2003). Multisource feedback in the assessment of physician competencies. *J Contin Educ Health Prof* 23(1): 4-12.

Miao, Y. & Koper, R. (2007). An Efficient and Flexible Technical Approach to Develop and Deliver Online Peer Assessment. Clark A. Chinn, G. Erkens, S. Puntambekar (Ed.) *Mice, Minds, and Society*, Proceedings of the 7th Computer Supported Collaborative Learning (CSCL 2007) conference, p. 502-511, July 16-21, 2007, New Jersey, USA.

Miao, Y., Sloep, P., & Koper, R. (2008) Modeling Units of Assessment for Sharing Assessment Process Information: towards an Assessment Process Specification, Proceedings of the 7th International

4.2 Self assessment tests for the Personal Competence Manager

As Miao points out in the chapter above, self assessment has by far the highest priority in the literature surveyed. This is line with the high priority given in this final reporting period to the completion of Task Force 7, responsible for integrating QTI tests into the Personal Competence Manager (situated in task 6.1) This involved the adaptation and extension of the WP6 infrastructure for QTI so that it could provide competence self-assessment tests within the Personal Competence Manager.

In terms of the analysis in the published chapter above we note that self assessment is a relatively simple form of assessment, and that it is often used as a preparation for other types of assessment, for example 360 degrees. It was therefore sensible to use similarly simple methods to support self assessment, making use of the additional flexibility which they offer. This was the logic of implementing self assessment as portlets within the Personal Competence Manager.

4.2.1 Take QTI test portlets

Users of the Personal Competence Manager have to provide information about themselves and the competences which they have achieved before they can be given guidance by the system about their best path towards achieving their competence development goals. They can do this by indicating their level of competence on a scale. In some cases learners are well aware of their competence levels, or may have taken prior courses in the system or have qualifications which clarify this, but in others they may well overestimate or underestimate this substantially. Consequently there is a need for a simple self-test functionality which can assist them in assessing their level of competence. The QTI editor in ReCourse could have been used to create such tests, but this would have required the use of a desktop application, to create and upload the tests, and then a publishing process on the server. A simpler solution was required, and so portlets were created with which teachers and others could create simple self-assessment tests to be made available to members of the community who sought to achieve a particular competence.

In this way simple assessments can be provided within the Personal Competence Manager, in the same way as simple learning activities can be provided within the system. These can be authored simply and conveniently as required. which could create and deliver simple tests from within the Personal Competence Manager, ensuring that the user can make an informed judgment about the competence levels which they should ascribe to themselves. Using the Learning Design Toolkit more sophisticated assessments can be developed using ReCourse, uploaded to the PCM, provisioned for groups of learners, and made available in Learning Paths.

This work was carried out within work package 6, but the of authoring and testing functionality is an integral part of the Personal Competence Manager, and is delivered as part of that system in WP3.

Extensive work was required to ready the NewAPIS runtime system for use with the PCM. This involved decoupling NewAPIS from the LD runtime systems with which it had previously been bundled. A “miniserver” was developed which can list and serve xml files stored in a folder, publish tests, and can handle multiple connections

and distinguishes users by their IP. All the user management and the database was shared with Liferay. An adaptation of all the tablenamees and sql syntax was done in order to make the package compatible with mysql. An interface was designed in order to handle the output of tests, and the input interface was also modified to deliver the XMLs of the QTI questionnaires.

In related work a CSS was created for NewAPIS so that the look and feel of the output of newAPIS could closely match that of the Liferay based PCM. Accordingly the render functions of newAPIS package were cleaned out of colours and layout information. Other changes were made to the distribution of content on the screen, and new tables and subsections added for distributing graphical information.

4.3 Assessment templates

In the chapter included above in section 4.1, the characteristics of four forms of assessment which are currently receiving substantial interest were systematically analysed from the perspective of process orchestration technologies. This was done by taking the first ten articles found in both databases, most of which provided examples of assessment types. These served as the input for the descriptions of the four forms of assessment.

This was done in order to

- a) confirm that the Learning Design Toolkit would be able to express these forms of assessment
- b) in order to prepare for exemplifying them with example UOLs
- c) identify specific modelling problems and propose solutions.

This work informed the development of templates for the Learning Design Toolkit, and the following templates have been prepared:

- 360 degree
- Peer assessment
- Formative assessment
- Summative assessment
- Adaptive learning based on QTI test
- Learning Outcomes negotiation.

More details about these templates, and others included in the ReCourse Learning Design Editor, are available in Appendix

The templates are available for download from DSpace at

<http://hdl.handle.net/1820/2283>

Additionally an example was produced which models a real-world performance assessment, in which three assessors use the same assessment form to evaluate three performances (playing violin) of a candidate. The final score is a weighted mean of all evaluations¹¹.

¹¹ <http://hdl.handle.net/1820/2248>

4.4 Running QTI assessments in the LD Toolkit

In this final reporting period the part of the NewAPIS code which contains information about the outcome variables has been more closely integrated with SLeD through the integration layer of Coppercore, known as CSSI (CopperCore Service Integration). These variables contain the information managed in the QTI test (such as values of the final score of a test, score of each individual items, number of correct responses, etc.¹² The reason for improving the management of these variables was twofold:

1) Enabling the management of level B properties between NewAPIS (IMS QTI) and Coppercore (IMS LD). One of the key ways in which IMS QTI and IMS LD can work together is to enable the results of tests to determine the flow of learning activities in an adaptive structure, indeed this was the main use of such tests in OUNL EML before it was disaggregated to create IMS LD. For this to work correctly the *outcome variable properties* created when a learner carries out a test have to be passed to CopperCore, and be associated with *level B Properties* in IMS LD. A system for achieving this has been implemented, creating a notification routine in APIS that sent the variables to an existing property notification mechanism in CopperCore

Following the schema provided by Vogten¹³ (Figure 6) Function A (property event) has been implemented in the CCSI package. This function is the one that delivers the outcome variables to the dispatcher, then the dispatcher gives the outcome to CopperCore (The client) (see function B). An important detail to take into account is that the declaration of the types of these variables in QTI must use *float* as the type for score. In CopperCore, on the other hand, properties cannot be defined as “floats”. In view of this a new function/tag <floatToInteger> has been provided in newAPIS to allow type changing in expressions, so that the two specifications can work together. This tag did not exist and now allows the system to work with integers and floats in outcome processing. All outcome variables are now provided to CopperCore in order to set them as properties. If they are not declared as properties in IMS LD CopperCore just ignores them.

¹² see more information in:
http://www.imsglobal.org/question/ktiv2p1pd2/imsqti_infov2p1pd2.html#section10059

¹³ Hubert, V. (2008) *Design and Implementation Strategies for IMS Learning Design*. SIKS Dissertation Series No. 2008-27. <http://hdl.handle.net/1820/1530>

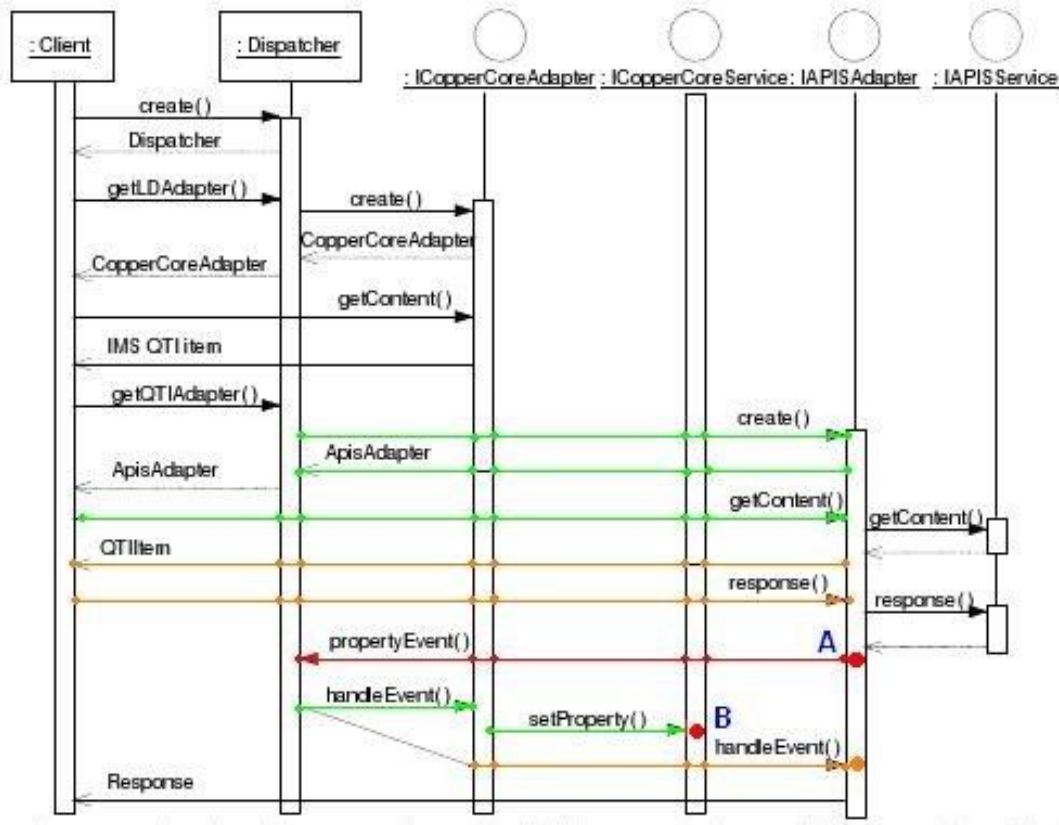


Figure 6. Management of outcome variables

However, fully integrating QTI raised some issues for the learning design player. In the standalone version and with ccsi running on the same server machine as player delivery server there were no problems. At the point where the player (SLED initially) was integrated with the portlet environment (Liferay) the runtime engine needed to be separated and run on a different machine. This meant that instead of direct (Java) calls between the player and the runtime engine web service calls needed to be used. Some work was required on the player side to implement these web services properly for APIS. The web service calls worked well for LD calls to coppercore but the APIS calls were not defined. These definitions were put in place allowing the SLED player to make proper use of APIS as a web service.

(2) Verifying if the outcome variables can handle new variables declared in the xml such as "EQFLEVEL". In the present release these outcome variables are well calculated in NewAPIS and are sent correctly to CopperCore.

In order to declare a new outcome variable it is necessary to add a line like to the xml of the test, as follows:


```
<outcomeDeclaration identifier="MY_SCORE_PERCENT"
cardinality="single" baseType="integer" />
```

Then explain how it will be calculated

```
</setOutcomeValue> <setOutcomeValue
identifier="MY_SCORE_PERCENT"> <product> <baseValue
baseType="float">100.0</baseValue> <divide> <numberCorrect
/> <numberPresented /> </divide> </product>
</setOutcomeValue>
```

The xml from the two TENCompetence QTI editors was tested with the current newAPIS implementation to establish that the interoperability functionalities were working.

5 Learning Design Runtime

5.1 Astro Learning Design player

When learners or teachers work with a Unit of Learning delivered by IMS Learning Design their activities and state are managed by a *Learning Design Engine*, in our case CopperCore. This ensures that the right activities, resources and services are delivered to the right person at the right time. However the Learning Design Engine has no user interface, it only returns values relating to the learner. Before teachers and learners can interact with the system these values have to be used to drive a user interface which represents the Unit of Learning. This is task of a *Learning Design player*.

The players available to TENCompetence, and other users of IMS LD, were, however, rather limited. SLeD, the best available Learning Design player and the one used by the project in pilots, is tied to a frame division of the screen and a tree metaphor for representing the Unit of Learning. It is also the result of extensive patching of what was originally a research tool rather than a delivery system, and has quality and maintainability problems. Consequently in DIP 4 it was decided that it was necessary to build a new player which could make use of up-to-date web programming techniques, and provide an alternative representation of UOLs. This is the Astro player, delivered in this report, which we now describe.

5.1.1 Introduction

The ethos behind the Astro Learning Design Player, was to present a different user interface to that provided in previous Learning Design Players. Typically such players have tended to show sometimes complex Learning Designs, using a tree view. This is a typical way to implement an xml binding into a user interface, as they tend to map easily together. However, as IMS Learning Designs can be complex, the tree view can become cluttered and unintuitive. Users will often be confused by the structure of a Learning Design and not understand how and why certain parts of the structure are there. The idea with Astro was always to be ‘activity’ centric. Most users do not care about high level abstractions in Learning Design, they simply want to complete the activities. To model this we decided to also provide a new linear way of show activities, using a filmstrip.

In essence the filmstrip shows each leaf node of the Learning Design, or more specifically it just shows activities. An activity is one of either a Support Activity, Learning Activity or an associated Environment. The user is able to navigate through the Learning Design using this filmstrip, by clicking on an icon representing one of the above activity types. Once a user has clicked on an activity, one or more sub windows will appear, showing the content related to that activity. The content can consist of, for example, learning objectives, prerequisites or services. Each of the sub windows is able to be minimized, maximized, dragged and resized. We wanted to implement this behaviour using sub windows, so that a user could make the most of the room available on screen. When the user does not want to look at a certain screen asset, they can simply minimize that window and then maximize the content they do

want to view. All user interface features in Astro are either collapsible or have the ability to be hidden, much more in thinking with a computer desktop.

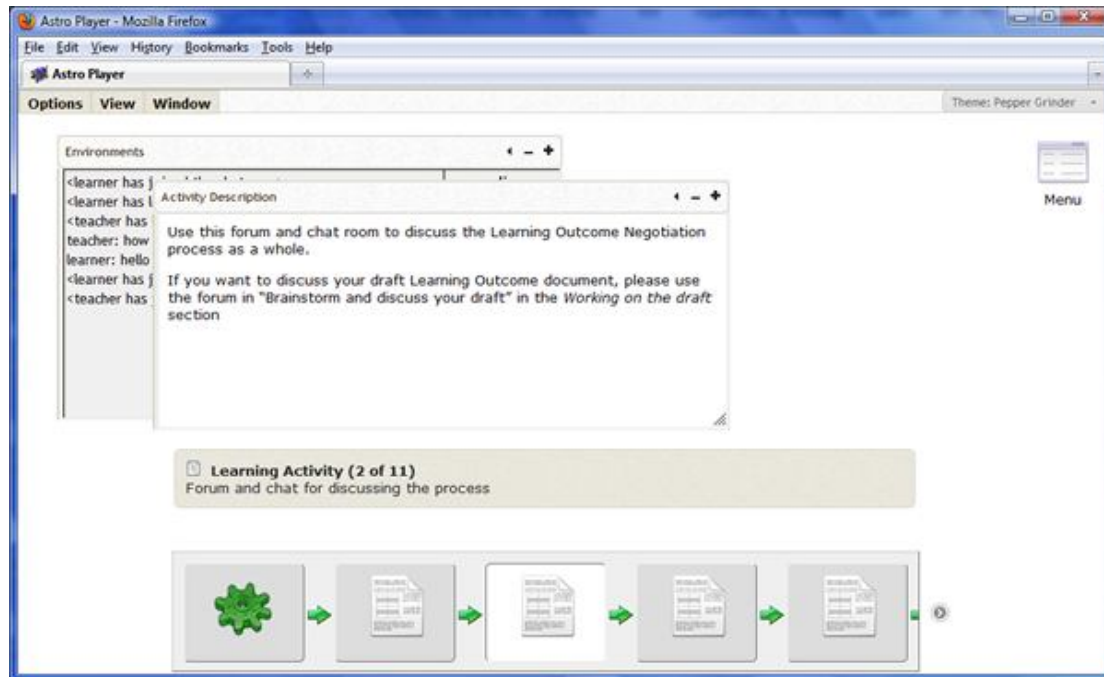


Figure 7: Astro, showing the use of windows

5.1.2 Implementation

We wanted to be able to show the user where they were in a currently loaded Learning Design. This means that unlike other players, we wanted to be able to mark activities as stages in the whole Learning Design. Ultimately this means we needed the ability to mark an activity as “Activity 2 of 12”, for example, where there are 12 activities and the currently selected activity is the second one. The problem with doing this is that you need access to the whole of the Learning Design structure to build this picture. Astro consumes the CopperCore¹⁴[1] service API to model Learning Designs, much like SLeD had done previously. However, CopperCore is a state machine. This means that CopperCore will only provide information on a given Learning Design's current state for a user. Typically this means the user sees just a snippet of the complete Learning Design. Future events are hidden completely. To mark the activities in stages, we needed the whole of the Learning Design. To do this, we query the CopperCore database directly. (the only time we bypass the CopperCore API). With this information we can build a picture of the Learning Design as a whole in its initial state. This is modelled as a set of java objects in memory. Next we use the CopperCore API to get the current state of a given Learning Design for a user. We then compare our original model with the state information, marking anything found in the state information as visible in the user interface in Astro. We now have a way of marking the activities in stages, but also can show future times as “locked” in the user interface, which means although the user can not complete these activities yet, they

¹⁴ <http://coppercore.sourceforge.net/>

are aware of them. These “locked” activities will only be unlocked once the CopperCore state machine reports they are available.

One of the design considerations when planning Astro, was the ability of the new player to leverage the use of Ajax functionality. For the uninitiated, Ajax allows a request for content in a browser to be loaded, without having to reload the whole of the current page currently visible. This means that content can be obtained from the server in the background and presented to the user by updating the browser's Document Object Model (or dynamic HTML). As the Astro server back end is written using Java servlets¹⁵ and the Learning Design is modelled using Java objects, we chose to use Direct Web Remoting¹⁶ for Ajax functionality. This library allows us to map Java objects on the server, to Javascript objects in the browser.

Another consideration in designing the user interface was whether or not to use an existing Javascript toolkit or not. There are many toolkits available which provide rich user interface features, such as custom dialogs, widgets and themes. One of the more popular toolkits around is JQuery¹⁷ and this is the one we chose to use. It is well documented, widely used and provides many features. Extensive use of JQuery is made in the Astro user interface. The tree view and filmstrip for example, both make use of JQuery functionality. JQuery does provide windowing functionality, however it does not currently support windows which can be easily maximized and minimized. To get around this problem we used the Fenster windowing toolkit¹⁸, which does support these features. The Fenster windows are styled so that they look like a JQuery type window. (using JQuery UI css).

5.1.3 The astro sequence of events

The user points his/her browser to the correct URL where Astro can be accessed. They are then prompted to login. Once the username has been entered, an Ajax request is made in the background to the Astro DWR servlet. This then makes a call to the CopperCore service API. The result of this determines if the user is valid and exists in the CopperCore database. If the user is valid, then the resulting step is to query CopperCore again for all available Learning Designs and return that this particular user is assigned to. CopperCore returns an xml fragment, which is then turned into Java objects by Astro. These are in turn passed back to the browser which are now available as Javascript objects. Astro now shows a dialog box with this information. The user can now select a particular run to show. Again, an Ajax request in the background is sent to the DWR servlet. This in turn queries CopperCore twice. The first time, the CopperCore database is queried to obtain the original Learning Design structure for this particular user. The result from CopperCore is an xml string, which is then converted into a Java model. (Note that at this stage, each of the items found in the structure are marked as invisible in the model). Next another request is made, this time to the CopperCore API, which provides the current state information. Again a

¹⁵ <http://java.sun.com/products/servlet/>

¹⁶ <http://directwebremoting.org/dwr/>

¹⁷ <http://jquery.com/>

¹⁸ <http://cross-browser.com/>

Java model is made. The first model and second model are compared. When an object exists in both of the models, it is flagged as available and so can be seen and accessed eventually in the user interface. Finally, the completed Learning Design model is passed back to the browser as a Javascript model. Next Astro will use this information to populate the Tree view and the Filmstrip view. The following diagram shows the layers involved in the architecture.

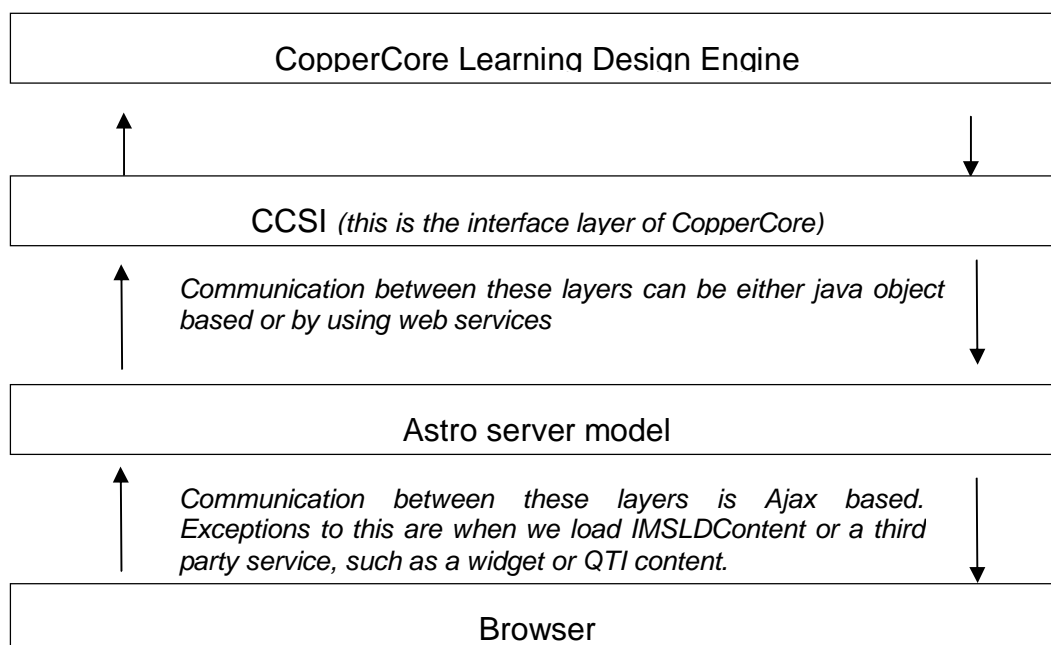


Figure 8: The architecture of Astro

5.1.4 User interaction

The quick start guide for Astro, provided in Appendix 2 provides a good guide to user interaction with the system.

5.2 Wookie

The Wookie Widget Server is the TENCompetence solution to the problem of providing flexible services for use with IMS LD Units of Learning at runtime. It was delivered in D 6.3, and work in this period has focused on promoting adoption, documentation, and successful submission of the application to the Apache Foundation Incubator.

Wookie was developed with strategy of enabling it to be reused in different contexts, so as to maximise potential adoption and support. Consequently it was designed with no dependencies on other TENCompetence services and tools. As a result it was possible to develop Wookie into a completely standalone application which could be integrated into a wide range of applications, for example Moodle and Elgg (as described in D6.3).

The success of this strategy, and also because it implemented the emerging W3C Widgets standard, led to Wookie being invited to submit to the Apache Software

Foundation (ASF) for incubation. This process was successfully completed in September 2009, and the Wookie incubator project at ASF is now actively developing the solution further, and attracting more contributors. Plugins for Wookie have been developed for Moodle, Wordpress, Sakai, LAMS, and Elgg – the latter three developed independently within their respective open source communities. The documentation produced by the project, including developers guides and FAQ, was adapted for submission to Apache and is now available at <http://incubator.apache.org/wookie/index.html> For the convenience of readers who would like more technical information about Wookie, including an introduction to widget technology, the developers guide submitted to Apache is included as Appendix 3 to this deliverable.

In addition Wookie has been reused by many other projects, and this has widened the pool of people contributing to the platform, and has added value to the TENCompetence project. For example, by extending the features of the widget server itself, by enabling integration of other products, and by improving the stability and performance of the widget server.

Since September the development work has been undertaken under the aegis of the Apache Wookie project and shared with a growing developer community. As of December 2009, Wookie had achieved 89% conformance to the W3C Widgets specification, and was one of three applications close to achieving full conformance.

5.2.1 Apache Software Foundation

In September 2009, Wookie acceptance into the Apache Incubator was formally confirmed. The incubator provides a mechanism for developing communities around open source software; once these become sustainable the project graduates to become one of the core Apache Software Foundation projects.

Since entering the incubator, a number of enhancements have been made to Wookie as part of its transition to a sustainable community project. These include a much improved approach to manage the software code itself, including dynamic management of dependencies, automated build and deploy.

One of the key requirements of an Apache project is ensuring that all the code is available under compatible open source licenses.

One of the first tasks in preparing the code was to check the ownership of every file used and obtain appropriate licenses and/or permissions. This required soliciting Contributor License Agreements (CLA) from all contributors, which included developers at the University of Bolton, Logica, and several external independent developers whose code was reused in Wookie. In addition, as the primary developers, the University of Bolton had to supply ASF with a corporate CLA.

Once all IP and rights were cleared, the software code was migrated to the ASF servers.

There was very positive reaction to the project, and in particular when we asked if we could credit the TENCompetence project and the EU funding, we received this response from Dan Brickley at W3C:

I'd love to see it included. The structure of academia tends to reward scholarly paper-publishing but doesn't really know what to do with software and data work. European projects also tend towards producing deliverables that are mostly likely to be giant PDFs rather than running re-usable code. So when we do finally get useful outputs from European research funding that enrich the open standards / open source scene, please let's not be shy in celebrating that! Maybe others will follow the great example, and start thinking more seriously about open source life-after-funding for their codebases, rather than taking a "throw the code over the wall and hope for the best" approach. Millions of euros get spend every year on these EU research projects, that's a lot of lines of code that could be going into the common pool...

Today Wookie makes use of the ASF services for code management, issue tracking, mailing lists, wik and website support¹⁹. This significantly reduces the overhead in continuing the project. More significantly, a team of experienced ASF members are mentoring the project to ensure its future success. Apache has also organised two meetings on Wookie, one in London and one at ApacheCon in Oakland California. The latter attracted sponsorship from LinkedIn.

5.2.2 W3C

The Wookie widget server is based on implementations of the W3C Widgets family of specifications, particularly Packaging and Configuration²⁰ and The Widget Interface²¹. As of December 2009, Wookie had achieved tested conformance to 89% of the specification according to W3C, and expected to reach 100% conformance by early 2010.

The project was able to contribute actively to the W3C standards process; this included participation in mailing lists, presentations at the W3C face-to-face working group meeting at Orange/France Telecom, and submitting a joint white paper with the PALETTE project²².

As a result of the contributions made to the specification as part of his TENCompetence work on Wookie, Scott Wilson from the University of Bolton was invited to become a member of the W3C Web & Hypertext Applications Group and to contribute formally to the development of the specifications.

¹⁹ Apache Wookie: <http://incubator.apache.org/wookie>

²⁰ Widgets 1.0: The widget Interface. M. Cáceres, R. Berjon, and A. Bersvendsen. W3C Working Draft 17 November 2009 (Work in progress).

²¹ Widgets 1.0: The widget Interface. M. Cáceres, R. Berjon, and A. Bersvendsen. W3C Working Draft 17 November 2009 (Work in progress).

²² Wilson, S., Bogaerts, J. and Sire, S. "Implementing the W3C Widget Specification in web containers: a comparison of Palette and Wookie". <http://groups.google.com/group/talk-about-widgets/web/implementating-the-w3c-widget-specification>

5.2.3 BONDI

Recent work has included support for additional specifications; This included implementing part of the BONDI device API specifications²³, enabling support for device hardware such as cameras and microphones, This enables authors to create widgets that take photographs using simple commands such as:

```
var camera = bondi.camera.getCameras()[0];
camera.takePicture(myfunction(pic));
```

Figure 9 shows this running in a Wookie Widget in Moodle.



Figure 9: Wookie Widget accessing laptop webcam using BONDI APIs

5.2.4 IMS Basic LTI , Desire2Learn and Blackboard

The team also implemented the IMS Basic LTI specification with the assistance of Chuck Severance at the University of Michigan²⁴. As a result of the latter work, it was possible to integrate Wookie with additional proprietary LMS platforms including Blackboard, WebCT Vista, and Desire2Learn (see Figure 10).

²³ See <http://bondi.omtp.org/default.aspx>

²⁴ <http://www.dr-chuck.com/csev-blog/000672.html>

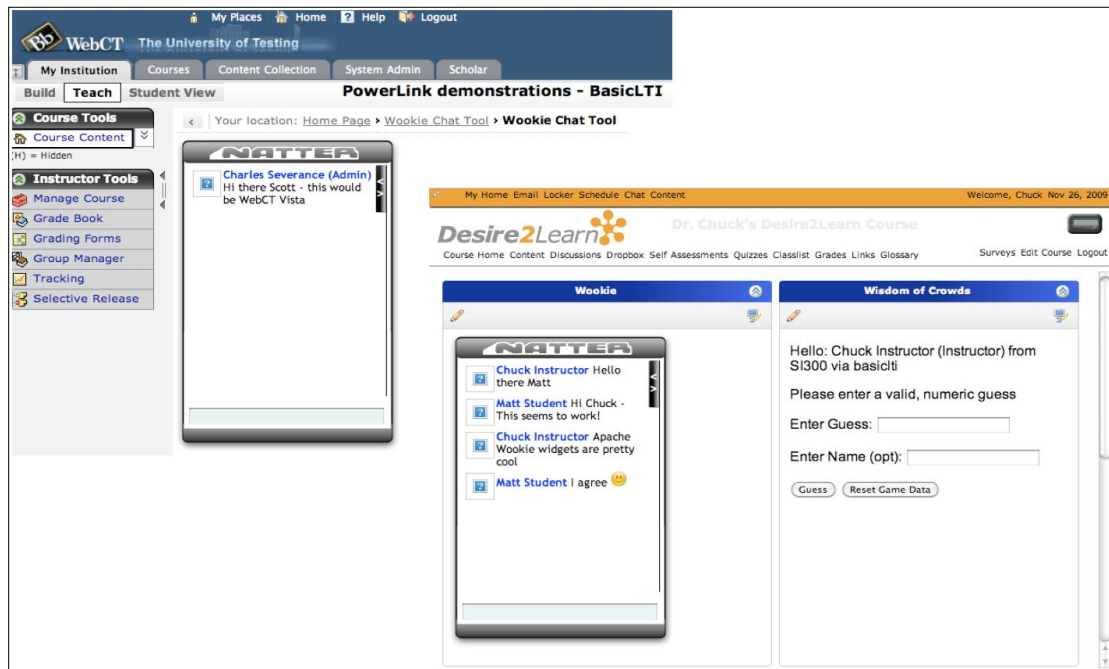


Figure 10: Wookie running in WebCT and Desire2Learn

In addition, the LAMS, Moodle, and Elgg communities created Wookie plugins based on the Wookie REST API²⁵.

5.2.5 Interaction with other funded projects

5.2.5.1 PALETTE

The PALETTE team also implemented their own implementation of the W3C Widgets specification, and there has been strong collaboration between the Wookie team and PALETTE as a result, including a joint white paper presented to W3C²⁶, and both projects contributing to a shared space for developing new ideas.

5.2.5.2 ROLE

The ROLE project has engaged with Wookie through its workshops on Mashups and Personal Learning Environments (MUPPLE) which have included Wookie demonstrations and papers. They have also used Wookie as part of their research outcomes, developing models incorporating analyses of Wookie's affordances²⁷.

²⁵ Apache Wookie REST API: <http://incubator.apache.org/wookie/wookie-rest-api.html>

²⁶ Wilson, S., Bogaerts, J. and Sire, S. "Implementing the W3C Widget Specification in web containers: a comparison of Palette and Wookie". <http://groups.google.com/group/talk-about-widgets/web/implementating-the-w3c-widget-specification>

²⁷ Palmer, M., Sire, S., Bogdanov, E., Gillet, D., Wild, F. "Mapping Web Personal Learning Environments". Proceedings of the Second International Workshop on Mashup Personal Learning Environments (MUPPLE09) Nice, France, September 29, 2009. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-506/palmer.pdf>

5.2.5.3 LTFL

LTFL has been working on extending Wookie to make it part of its solution for supporting social and informal learning.

5.2.5.4 MATURE

The MATURE project implemented Wookie as part of their portal solution, and extended the platform with additional functionality²⁸. The University of Paderborn are working with the Apache Wookie project on developing their extensions for inclusion in the core project.

5.2.5.5 ICOPER

ICOPER has investigated the use of Wookie for both its internal organisation and also as part of its analysis of standards and technologies in TEL. The project invited the Wookie team to a Widgets Workshop hosted by the project.

5.2.5.6 New proposals

Four consortia developing new project proposals involving Wookie have also contacted the project team. Wookie is also the focus for a new funding programme from the UK's JISC funding agency, with a number of projects to be funded developing and piloting the use of widgets, and developing Wookie plugins for additional platforms.

5.2.5.7 "Talk About Widgets"

This group²⁹ was originally set up by the PALETTE project (see above) but has become a useful space for discussing advanced concepts and new functionality for widgets. Most recently this has focussed on developing specifications for inter-widget communication drawing on research undertaken in PALETTE, ROLE, MATURE, and LTFL. This group looks set to remain an important forum for research in this area long after the conclusion of the current projects.

5.2.6 Conclusions

This final stage of TENCompetence work on Wookie has been highly satisfactory, with acceptance of the code into the Apache Incubator, and a the high level of interest and adoption beyond the project. As a result we were able to progress the solution to an advanced stage, and have excellent prospects for future sustainability as a result of the efforts made by the project team at building an active developer community as well as code.

The fact that the solution which we developed for the Learning Design Toolit provides services through a generic widget infrastructure has proved very positive for the platform as a whole, and reduces the amount of LD-specific code that needs to be maintained as part of an LD runtime platform. Conversely, developing LD in a direction that converges with developments in the wider web (including both web 2.0

²⁸ Nelkner, T. (2009). "An Infrastructure for Intercommunication between Widgets in Personal Learning Environments". Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All; Second World Summit on the Knowledge Society, WSKS 2009, Chania, Crete, Greece, Sept 16-18, 2009. Proceedings <http://www.springerlink.com/content/wm2r08466j217835/>

²⁹ Talk About Widgets Google Group <http://groups.google.com/group/talk-about-widgets>

and mobile applications) should enhance the ability to deploy LD in more environments, and to both add value to and receive added value from external innovations, e.g. third-party widgets, and Wookie plugins.

6 Integration

Much of the work described in previous sections of the deliverable could be classified as “integration”, as that is a significant part of the development work carried out (for example Wookie, QTI in the PCM, and Learning Design). Indeed IMS Learning Design itself may be considered an orchestration mechanism. However those aspects are presented in the thematic sections which correspond to the main focus of the work. In this section we report on the development of portlets whose only purpose is integration.

As mentioned in the introduction, in a chapter by Griffiths and Liber³⁰ reporting on our work in TENCompetence, we describe how the IMS LD specification was developed with the idea that the landscape of eLearning would consist of a relatively large number of competing Virtual Learning Environments (VLE). It was (it seems mistakenly) assumed that the role of IMS LD would be to provide interoperability between VLEs. At present, however, we find that the sector is dominated by one very large commercial vendor, and a few Open Source systems, led by Moodle. At the same time the VLE is being questioned, as a historically necessary but transient type of application, for example in the Personal Learning Environment proposed by Liber³¹, which shares much of its approach with the Personal Competence Manager.

From this perspective it can be seen that the integration of the Learning Design Toolkit into the Personal Competence Manager provides a demonstration of how learning activities can be disaggregated from the Learning Management System. The loose coupling between the Learning Design Toolkit and the PCM means that they can be installed independently on different servers and, potentially, the same learning activities can be delivered in different contexts.

To achieve this the following three portlets have been created:

The Learning Design Administration portlet

This enables Liferay users in the Manager/Teacher role to carry out management functions in publishing UOLs, setting up runs (instances) of those UOLs, and managing users. The portlet provides a view onto all the UOLs published on the CopperCore server associated with the PCM installation, and a means of publishing new UOLs. This can be done directly from the desktop, or by using the 'Search Fedora repository' functionality to search for and upload IMS LD packages. Once a UOL has been published, runs can be created. Clicking any of the available runs lead to a page with list of enrolled users and option to add new users from the system, and groups defined on the Liferay system can be added.

³⁰ Griffiths, D. and O. Liber, Opportunities Achievements and Prospects for use of IMS LD, in *Handbook of Research on Learning Designs and Learning Objects*, L. Lockyer, et al., Editors. 2008, Information Science Reference.

³¹ See Interactive Learning Environments Special Issue: Narrative and Interactive Learning Environments, volume 16 Issue 3, 2008. Liber, Oleg & Johnson, Mark (Eds.)

The SLeD player portlet

This portlet enables a Learning Design activity to be accessed transparently from within the Personal Competence Manager. It accesses the same CopperCore Learning Design Engine as the LD Administration Portlet, and delivers the activities which have been published there to the learners who have been enrolled on them. This involved substantial work on the SLeD player, specifically

1. Convert all old (unmaintained) struts 1.1 library based action classes to utilise Struts2 library. Also create struts configuration files.
2. Create servlets for common server interactions (player and qti) to make use of AJAX. Ajax based calls eliminated the need for reloading the entire page (thereby faster).
3. Change all the XSL files used in SLED portlet to amend the generated html output to one compatible with portlets.
4. Modify the Javascript code/eliminate some libraries (dtree, prototype etc) to make sled work within the portal environment which uses jQuery library.
5. Create javascript routines for use within the portlet page in response to user interactions.
6. Modify sled code to utilise the logged in liferay user's credentials while providing access to runs.

The Link Tool portlet

The Link Tool enables learners to publish and access UoLs. As discussed in the paper below, the tool includes a registration module in the main page that a users access when entering for the first time. Once logged in, the user is shown a view that includes a subset of those reserved for the administrator role in the LDRuntime environment. This enables the learner to link to a particular run, and perhaps create a run if one did not exist for that UOL.

Testing

In addition to the evaluation of the link tool described in the paper included in the following chapter, the portlets were tested by being used in the eLearning Reference Implementation of the Personal Competence Manager, which was demonstrated at the Manchester Open Workshop. The performed as expected, although some problems with the use of web services and QTI were exposed which have since been addressed.

7 Evaluation

7.1 Evaluation of the LD Toolkit

This evaluation action added to previous evaluations by using the integrated LD Toolkit in a near final version (integrated LD QTI authoring, integrated LD and QTI runtime, publishing UOLs on the the PCM).

Earlier evaluation of ReCourse had provided authors with a short training input, after which they were able to create simple UOLs (see D6.2), and this was born out by a focus group with authors conducted in May 2009. In the present evaluation no such training was provided, in order to provide a more demanding test of usability which may be faced by many users outside TENCompetence. On the other hand the users involved had a stronger technical background than earlier evaluation actions.

Thus the objectives of the evaluation were

- a) to assess the use of the integrated LD Toolkit in an unsupported environment
- b) to identify any outstanding usability issues, assess their severity, and feed back to the development team for attention prior to the final release.

Participating authors found that they could create, publish and interact with the UoLs successfully, as summarised by one participant who said: *“Both ReCourse and the QTI editor really facilitate the creation of UoLs and QTI tests by avoiding the tedious work of dealing directly with the specification”*.

In general the user interface of ReCourse interface was very well received, and it helped in managing the tasks involved in creating a UOL.

Some issues were identified with functionality and interface, but participants commented that the majority were not critical. The only issue shared by a number of participants was that the development team might consider separating *roles* and *activities* onto different buttons.

All participants commented that in some areas these tools used rather technical language. This is not necessarily a problem given that some functionality is intended for advanced users. However, it seems that it would be useful to expand explanations about terminology (or change them where appropriate) and provide more guides for beginners. One participant suggested adding more templates of UoLs³².

The detailed feedback on user interface and functionality provided input to the preparation of the final release of the Learning Design Toolkit. All the major usability and functionality problems identified, which largely concerned QTI aspects, have been addressed in the final release of the software.

7.2 Included paper: Evaluation of the Link Tool Portlet

The Learning Design integration portlets were evaluated in a pilot, with a particular focus on the Link Tool. The results were delivered in a paper delivered to the final

³² In the version of the LD Toolkit used in this evaluation the full set of TENCompetence templates was not available

TENCompetence workshop in Manchester and accepted for publication. This section is drawn from a pre-print that paper.

A tool for the auto-management of Units of Learning: the Linktool

Mar Pérez-Sanagustín, Roy Cherian, Davinia Hernández-Leo, David Griffiths and Josep Blat

Abstract — *Prior to learning with an IMS LD Unit of Learning it is necessary that a run is created, learners added, and a link to the run provided. In the context of LifeLong Learning or Open Learning Resources there may be no designated administrator available to control the number of users that are going to participate in a UoL at any given time, or to provide links to the appropriate active run. In this case the user has to take responsibility for this process before they can start on the learning activities. There is therefore a need for new tools supporting the auto-management of UoLs conforming to the IMS LD specification. This paper presents a list of requirements arising from using Units of Learning in a pilot performed in the context of La Verneda School for adults in the Agora association. We also present the first version of the tool developed according to the requirements and a summary of the results of using it in a second pilot in the same school.*

1. INTRODUCTION

A 'unit of learning' is an abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc [6]. When these units of learning are computationally represented following the IMS LD specification they are called a full ready-to-run Unit of Learning (UoL) that can be authored, interpreted and run in different tools. The IMS LD is an educational modeling language used to develop applications in educational contexts [6, 8]. In the TENCompetence project, this specification has been adopted as the facto standard.

One of the commitments of the project is the adoption of the IMS LD specification in diverse and real educational contexts. Nowadays, different tools conform with IMS LD for supporting the edition [2, 4], the instantiation [3] and runtime [1] have been developed. Some of these tools have been successfully tested in different learning contexts [5]. However, in a lifelong learning context these tools show some limitations. A pilot study carried out in the Association of Participants Àgora [9], there were identified a set of problems regarding the management of UoLs. To execute and perform a UoL it is necessary to define the list of students in the course, the roles that each student is going to play and the course that they participate. For that, it is necessary to create in the production time what is called a run. A run is an instance of a course that serves as a bridge towards the students and a UoL [12]. Only those students registered in a run are allowed to execute (or run) the UoL. Normally, the creation of a run, and the assignation of the students to it is done by an administrator or a teacher that decides who is participating in which UoL. But in a lifelong learning context each learner organizes his/her own training and decides in which course he/she wants to participate. The figure of a teacher disappears and the learners should be able to create their own runs and manage his/her courses. The experience of the pilot showed that the current tooling is not enough for supporting the self-management of UoLs in Lifelong learning contexts and new solutions are needed. Different solutions addressing similar problems have been developed in other areas,

e.g. in identity roll-on and roll-off processes in a company or automatic subscriptions to online course with Moodle Platform [10]. Although these solutions are enough for covering the necessities in their particular areas, they are not enough for solving the requirements in a scenario involving auto-management of UoLs a Lifelong Learning context. Moreover, it does not exist tooling compliant with IMS LD supporting the user in the management of their courses.

The aim of this study is to understand the needs of the UoL's management in Lifelong Learning contexts. More concretely, we analyze the use of LD Runtime System in the context of Agora and propose a tool called LD admin or Link Tool as a solution to support the auto-management of UoLs in such as contexts.

II REQUIREMENTS FOR SUPPORTING THE SELF-MANAGEMENT OF UOLS IN LIFELONG LEARNING CONTEXTS

To understand the requirements arising from the management of UoLs in a Lifelong learning context, we analyze a realistic experience of a pilot carried out in the Association of Participants Àgora. The main aim of the pilot was to implement, test and investigate the benefits of the TENCompetence infrastructure and its support for the participants' competence development [9]. The participants in the pilot used the Personal Development Plan (PDP), the PDP tool developed in the TENCompetence project, as the central tool for the creation of their own personal development plans and the performance of the activities. The needs of the UoL's management in Lifelong Learning contexts. More concretely, we analyze the use of LD Runtime System in the context of Agora and propose a tool called LD admin or Link Tool as a solution to support the auto-management of UoLs in such as contexts.

Some of the activities were UoLs codified in IMS LD that run in the TENCompetence LD Runtime System [1]. The LD runtime system is compliant with the IMS Learning Design specification [7] and facilitate Some of the activities were UoLs codified in IMS LD that run in the TENCompetence LD Runtime System [1]. The LD runtime system is compliant with the IMS Learning Design specification [7] and facilitates the provision of structured activities (similar to courses) that learners can follow as part of their competence development. It provides an administrator view for uploading UoLs, registering users and creating runs and a player view that allows the learner to perform the activity. In this section we describe the use of the LD Runtime's functionalities for managing UoLs in this context and report and analyze the solutions adopted to deal with its limitations. The results from the analysis lead to a set of requirements needed for the development of a new tool able to support the whole process in such as type of contexts.

A Administrating UoLs in the 1st Agora Pilot

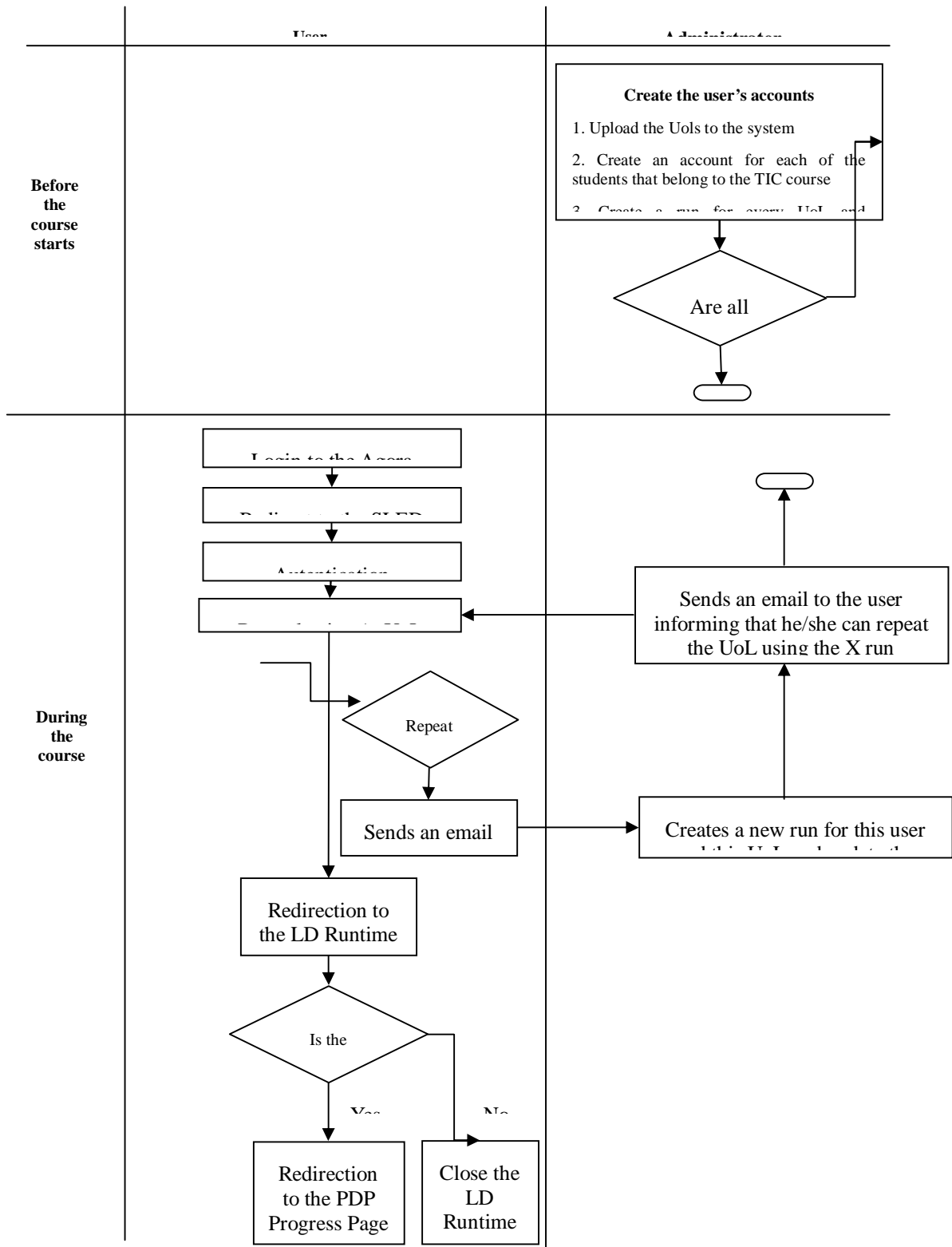
All the UoLs in the pilot were created with the LD Editor Recourse [2] by two experts. All were designed including collaborative widgets (chats in most of the cases) and with a unique learner role (UoLs can be designed including different roles with different privileges when performing the course). The administrator of the system (an IMS LD expert) used the LD Runtime administration section for uploading the UoLs to a server and registering the users participating in the experience. The administrator generated one instance for each course by creating a unique run for each of the UoLs uploaded and enrolled all the participants to these runs for associating the participants

registered to the courses available. All these steps were done before starting the course.

When the pilot started, the learners accessed to the UoLs from an activity in the PDP in which there was a link to the LD Runtime and the title of the Unit of Learning corresponding to the activity. When clicking on the link, they were redirected to the LD Runtime (via an Internet browser), log in to the system with the credentials facilitated by the administrator and visualized the list of runs associated to the UoLs available in which they were previously registered. By clicking on the run, they could access to the LD Runtime players and run the UoL to perform the activity.

From the experience of the pilot, there were detected some problems and limitations of the LD Runtime system not only regarding the administration processes before the course but also in supporting unexpected situations once started. In the next lines, we describe the main problems and the solutions adopted. Figure 1 depicts the full workflow of the whole process.

- Registering the users. The LD Runtime does not include any registration module for allowing the users register the system on their own. For the pilot, the registration process was carried out by the administrator. Before the pilot started, all the users enrolled in the course were previously registered to the system and informed about their credentials via email. However, during the pilot some students that were not registered to the course joint it once it had started. As a solution, the administrator maintained a list of users with their respective users and passwords that was updated when a new user joint the course.
- Repeating a course. One of the main problems that appeared during the pilot occurred when some students wanted to repeat one of the courses that they had already performed. Since the instance of a course is related to a unique run, a user can perform the UoL only one time and all the actions are stored. Once the activity is finished, if the learner access again to the same run, they find it finished. For the pilot, this need was solved via email. When the users wanted to repeat a UoL they sent a message to the administrator and he/she created a new run of the same UoL only for that user. The administrator maintained a list with the runs created for each of the users and inform them when the run was ready.
- Identifying the correct UoL. The user linked the LD Runtime player from a particular activity in the PDP. Once in the player, they were showed a list of runs. But, from the list, the learner should choose only that run indicated in the activity of the PDP. The users had some difficulties in distinguishing the run they had to select. For facilitating this process, all the runs were named with the title of the UoL and the link in the PDP activities were edited exactly with the same name. In that way, the users could better identify the correct run. For avoiding conflicts with the name of the runs associated to the same UoL (see the case of repeating a course in the bullet above) all the runs were described with the number of the run that the user was going to play (e. g. “Practice Vocabulary-1” for the first run and “Practice Vocabulary-2” for the second run of the same UoL titled “Practice Vocabulary”).



Pérez-Sanagustín et al. Figure 1: Flow diagram of the whole process of following a course, from the PDP to the LD Runtime player, and the roles involved in the process.

B Analysis of the requirements

If we examine the whole process followed in the Agora Pilot, we observe that all the solutions adopted for managing the UoLs lead on the Administrator. However, in a Lifelong learning context the learners should be able to register to the system and create their own runs of those UoLs they are interested in. Therefore, it is necessary to provide the learner with administration tools. From a detailed analysis of the problems detected in the Agora pilot, we expose in Table I the limitations of the LD Runtime tool and propose a set of requirements as a basis for developing a tool for the auto-management of UoLs.

List of requirements detected from the Agora pilot

Limitations of the LD Runtime System	Requirement
<p><i>REGISTRATION PROCESS</i></p> <p>The LD Runtime does not provide any registration module for allowing the user to create an account for accessing the system. An administrator is needed for creating the user accounts.</p>	<p>R1. Provide the system with an automatic User Registration/Authentication Module</p>
<p><i>UPLOAD UOLS</i></p> <p>The LD Runtime only allows the administrator to upload UoLs.</p>	<p>R2. Provide the system with functionality for the learner to upload his/her own UoLs.</p> <p>R3. Add the possibility of deleting a UoL. For avoiding problems with user privileges this functionality should be restricted to the owner of the UoL.</p>
<p><i>MANAGING RUNS</i></p> <p>The runs for the UoLs are only managed by the administrator in the LD Runtime.</p>	<p>R4. The learner should be able to create their own runs for a particular UoL and to assign a starting date for accessing the course.</p> <p>R5. The learner should have the possibility of accessing to an already created run. This run can be previously created by a teacher/administrator/expert or another user of the system.</p> <p>R6. The learner should be able to delete his/her own runs.</p>

The problems in the above table arise directly from the necessities detected during the Agora Pilot and are very context-related. But, the requirements extracted are enough generic for covering the needs of other Lifelong Learning scenarios. In section IV we shed light over other necessities that could be also considered.

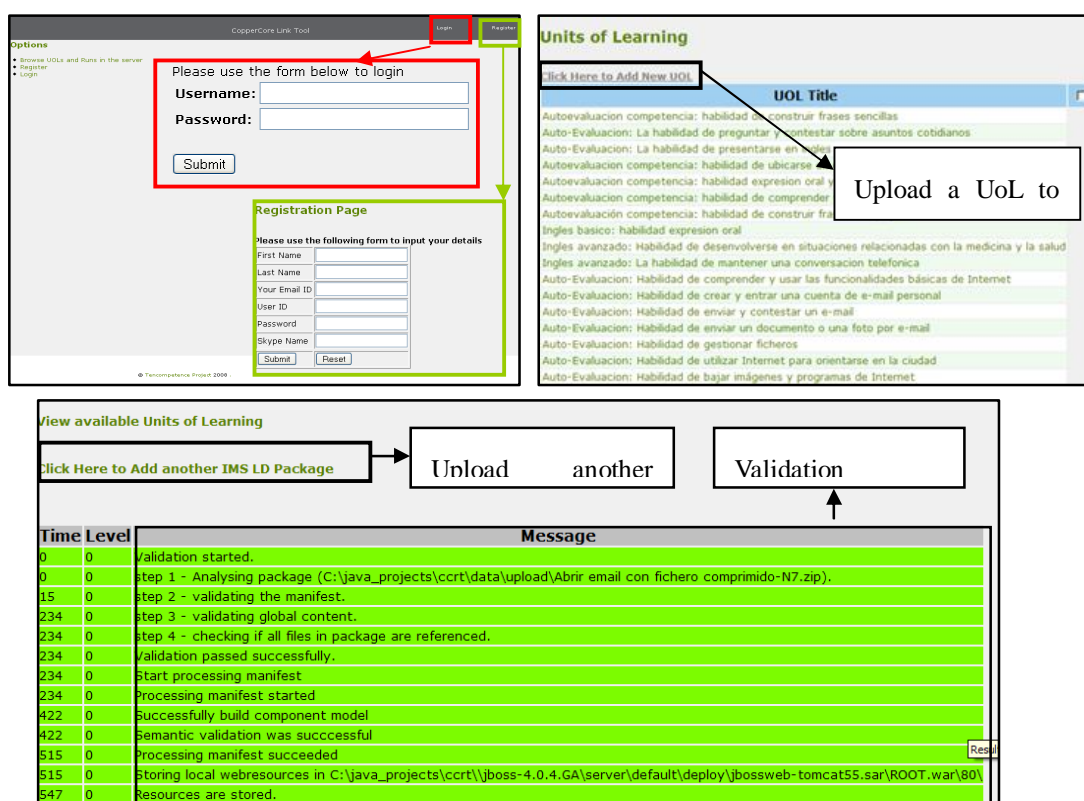
III. THE LINK TOOL FOR SUPPORTING THE AUTO-MANAGEMENT OF UOLS IN LIFELONG LEARNING CONTEXTS

This section presents a web tool called Link Tool as a first effort for supporting UoLs management in Lifelong Learning scenarios according to the requirements detected in

the first Agora Pilot. The firsts outcomes of using the Link Tool framed in a second pilot in the same school are also presented here.

A The Link Tool

The Link Tool is a web-based application that enables the auto-management of UoLs. According to requirement R1 in table I, the tool includes a registration module in the main page that a users access when entering for the first time. It also includes a section for login in case that the user is already registered (Figure 1 on the top left position). Once logged in, it is shown to the user a unique view that mixes some of the functionalities reserved for the administrator role in the LDRuntime environment with those reserved for the user. According to requirement R2, the user can upload his/her own UoLs (Figure 1 on the top right position). It is validated by the system and included to the list of UoLs available if it does not have errors (Figure 1 on the bottom). The user can also delete his/her own UoL if necessary (requirement R3).



Pérez-Sanagustín et al. Figure 2: Screen shots of the link tool. Top left: Main page with the registration and login menus. Top right: Page for uploading UoLs. Bottom: Validation page once the UoL is uploaded.

The Link tool allows the user to manage his/her runs. It offers the possibility of creating a run of an existing UoL (requirement R4) or joint an existing run (requirement R5) created by other user. This last functionality is especially useful in case of UoLs including some collaborative elements such as a chat or a forum. For example, if a user creates a UoL with a forum activity and a run associated to it, other users should be able to register to this run in order to visualize the messages in the forum. Otherwise, each user will belong to his/her run and will not visualize the messages of other learners. For that cases, it has been also included the possibility of

picking a date for accessing a run. This run can be only accessed in that date and, therefore, force the users to perform the course in that period. The owner of a run is able to delete it.

B Using the Link Tool in a LifeLong Learning context

To understand how the Link Tool helps in supporting the auto-management of UoLs in a Lifelong learning context, the tool was used in a second pilot carried out in the same Association of Participants Agora [11]. The aim of this second pilot was also to investigate the benefits of the TENCompetence infrastructure in the school. The main difference between the tooling used in the 1st and the 2nd pilots was that, in the 2nd one, the tools were improved: switched to a Web version and integrated in a LifeRay portal. The users accessed to this portal directly from the Web page of the association for registering to the system. As in the 1st pilot, the PDP was the central tool for generating the personal development plans and performing the activities. Some of these activities were UoLs. In this case, the UoLs were managed using the Link Tool instead of the administrator interface of LD runtime system.

Some of the UoLs used in this pilot were recovered from the previous one and new ones were created by two experts. However, in this case, each user could freely create his account using the registration module when entering the system for the first time.

The users accessed to the UoLs using a link (a URL pointing to Link tool) included in the description of the PDP activities. For that, the two experts created a default run for each of the UoLs uploaded by the coordinator starting the same day than the pilot started and link directly to that run from the PDP. The link directly redirects the users to this generic run and started following the course. In that way, the users did not have to search the correct UoL from the list of courses available and login to the system every time. It was also a good solution for those UoLs including collaborative modules such as a forum, in which different users should be included in the same run. When a user wanted to repeat a course, he/she could access the Link tool directly with login credentials, select the UoL he/she wanted to repeat and create his/her own run. This functionality avoided the necessity of maintaining a list of runs per user and facilitated very much the administration tasks of the course coordinator.

The “linking” URL used in the PDP tool could also contain information such the specific run to be used from among the number of runs available for a UoL as well as the user role for completing the run. This ensures better control in administrating UoL runs in more complex settings.

The results from the first experience show the Link tool as a good solution for supporting the administration of UoLs in a LifeLong learning context. On one hand, it provides the learners with some administration functionalities that allow them to freely manage their own courses. During the 2nd pilot, participants did not have to wait when they wanted to repeat a UoL because they could create his/her own runs. On the other hand, it facilitates the tasks of the course coordinators that only have to take care of making the UoLs available.

IV CONCLUSION AND FUTURE WORK

The existing IMS LD tooling fails in supporting the needs arising from the UoL's management in Lifelong Learning contexts. This paper presents the results of analyzing these necessities by studying the LD Admin IMS LD tool in the context of the Association of Participants Agora. The results of this study are a set of

requirements that have served as a basis for the development of the Link Tool. This new tool compatible with LD is an improvement of the LD Admin tool that provides the learners with some administration functionalities that allow them to manage their own courses. A first experience in the Agora school has shown the Link Tool as a good support for the auto-management of UoLs in Lifelong Learning contexts.

From the experience, there were collected some problems regarding the use of the Link tool. Most of them were related with the vocabulary used in the interface and usability aspects. Specific words such as run were not well understood by the users. In future versions of the tool, the vocabulary should be adapted to the context and the visual aspects of the interface improved.

Currently, the Link Tool is being adapted to be included as a portlet in a LifeRay portal so as to be integrated with other TENCompetence tools. A portlet is a module in a LifeRay portal that can be added or deleted according to the user needs. This new version of the Link Tool will maintain the main functionalities of the current version and improve some of the usability problems observed during the experience in Agora. For the integration, some of the modules such as the registration module are changing because all the users will register to the LifeRay portal and not to the tool itself. All this new functionalities will be tested in future experiences beyond the TENCompetence project for analyzing if the requirements covered by the Link Tool are still enough for the auto-management of UoLs in Lifelong Learning contexts.

ACKNOWLEDGEMENT

This work was supported in part by the European Commission in the TENCompetence project IST-2004-02787

REFERENCES

- [1] LD Runtime, *Website*. Available: <http://www.tencompetence.org/ldruntime/> [October, 2009]
- [2] Recourse, *Website*.. Available: <http://www.tencompetence.org/ldauthor/> [October, 2009]
- [3] Hernández-Gonzalo, J.A., Villasclaras-Fernández, E.D., Hernández-Leo, D., Asensio-Pérez, J.I. & Dimitriadis, Y. 2008, "InstanceCollage: A Graphical Tool for the Particularization of Role/Group Structures in Pattern-Based IMS-LD Collaborative Scripts", *Advanced Learning Technologies, IEEE International Conference on*, vol. 0, pp. 506-510.
- [4] Hernández-Leo, D., Villasclaras-Fernández, E.D., Jorrín-Abellán, I.M., Asensio-Pérez, J.I., Dimitriadis, Y., Ruiz-Requies, I. & Rubia-Avi, B. 2006, "Collage, a Collaborative Learning Design Editor Based on Patterns Special Issue on Learning Design", *Educational Technology & Society*, vol. 9, no. 1, pp. 58-71.
- [5] Hernández-Leo, D., Bote-Lorenzo, M., Asensio-Perez, J., Gomez-Sanchez, E., Villasclaras-Fernandez, E., Jorrin-Abellan, I. & Dimitriadis, Y. 2007, "Free-and Open-Source Software for a Course on Network Management: Authoring and Enactment of Scripts Based on Collaborative Learning Strategies", *IEEE Transactions on Education*, vol. 50, no. 4, pp. 292-301.
- [6] IMS LD 2005, *IMS Learning Design Best Practice and Implementation Guide, Version 1.0 - final specification, Website*. Available: http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html [October, 2009].
- [7] Koper, R. & Olivier, B. 2004, "Representing the learning design of units of learning", *Educational Technology & Society*, vol. 7, no. 3, pp. 97-111.
- [8] Koper, R. and Tattersall, C. 2005, *Learning Design: A Handbook on Modeling and Delivering Networked Education and Training*, Springer.

- [9] Louys, A., Hernández-Leo, D., Schoonenboom, J., Lemmers, R., & Pérez- Sanagustín, M. "Self-development of competences for social inclusion using the TENCompetence Infrastructure", *Journal of Educational Technology and Society*, vol. 12, no. 3, pp. 70-81.
- [10] Moodle, *Website*. Available: <http://moodle.org/> [November, 2009].
- [11] Santos, P., Llobet, W., Hernández-Leo, D. and Blat, J. (accepted) "*QTI for self-assessment and embedded- assessment in competence oriented scenarios: The Agora Case*", International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), 4-6 November, Barcelona, Spain.
- [12] Tattersall, C., Vogten, H., Brouns, F., Koper, R., Van Rosmalen, P., Sloep, P. & Van Bruggen, J. 2005, "How to create flexible runtime delivery of distance learning courses", *Journal of Educational Technology and Society*, vol. 8, no. 3, pp. 226.

8 Appendices

8.1 Appendix 1: List of Templates

The following templates have been developed for use with Recourse. Each template is accompanied by a description of how it works and by an UoL as an example of how this can be implemented.

Template	Brainstroming
Template file	Brainstorming_UoL-TEMPLATE.zip
Title of UoL example	Preparing a trip around Spain (Brainstorming)
Link to UoL example	http://www.tecn.upf.es/~daviniah/A1-ATripAroundSpain-Brainstorm.zip
Level of LD	Level A, includes forum, chat, vote
Use of IMS QTI	None
Educational technique	Brainstorming
Description of example UoL	Unit of Learning for basic learners of Spanish (A1). It aims at reinforcing the practice of using the vocabulary for shceduling events such as dates and seasons and for travelling. It includes chats and forum activities combined with a voting activity.
Authors	Mar Perez Sanagustin
References	Design derives from Prolix collection of pedagogic models Content adapted at http://www.onlinespanish.org/answers-crosswords.html
Description	The general idea of brainstorming is that the participants generate new ideas in a group effort in order to solve problems creatively
Category	Discussion-oriented models

Template	Guided discussion
Template file	Guided Discussion_UoL.zip
Title of UoL example	Give your opinion (Guided discussion)
Level LD	Level A
Use of IMS QTI (which question types?)	No
Educational technique	Guided discussion
Description of example UoL	Unit of Learning for Intermediate learners of Spanish (B2). It aims at making the students give their opinion about a news. They have to write their main opinions and discuss it with their colleagues.
Authors	Mar Pérez Sanagustín
References (Materials, pedagogical technique, template, etc...)	Content adapted at http://ec.europa.eu/spain/images/unds.didacticas/profe_libro_es.pdf El Pais: http://www.elpais.com/articulo/andalucia/Convivir/mimos/mendigos/elpepiespand/20091111elpand_10/Tes
Description	Trainer-guided discussions give learners a chance to develop critical thinking, clear oral expression, as well as experience in posing and responding to questions.

Category	Trainer-driven models
-----------------	-----------------------

Template	Peer Learning
Template file	Peer_Learning_UoL-TEMPLATE.zip
Title of UoL	Cooking
Level LD	Level A
Use of IMS QTITemplate file	None
Educational technique	Peer Learning
Description of the UoL	Unit of Learning for Intermediate learners of Spanish (B2). It aims at making the students understand the discourse of a native speaker. The students has to been able to summarize and explain the main contents of the video.
Authors	Mar Pérez Sanagustín
References (Materials, pedagogical technique, template, etc...)	Content adapted at: http://www.zappinternet.com/video/wigTluYweK/Como-preparar-Tortilla-de-Patatas You tube: http://www.youtube.com/watch?v=Oy5esAkpEk http://www.youtube.com/watch?v=KozLYSILHb4 http://www.youtube.com/watch?v=Oy5esAkpEk&feature=related
Description	Peer learning provides learners an opportunity to instruct each other on relevant topics. Learners experience higher self-efficacy, which has been shown to be positively related to performance in a task.
Category	Collaborative-Assessment-driven models

Template	Inquiry-based learning_UoL
Template file	Inquiry-based learning_UoL.zip
Title of UoL	Discuss bullfighting
Use of IMS LD	Level A, chat widget
Use of IMS QTI	None
Educational technique	Discussion
Description of the UoL	Unit of Learning for Intermediate learners of Spanish (B1). It aims at making the students investigate about Spanish cultural tradition, while they can visualize the diverse opinions claimed by Spanish people and associations. And finally to be able to defend her/his own opinion about it in a living chat room.
Authors Template file	Miguel Angel Carralero Jonathan Chacón
References	None
Description	The essence of inquiry-based learning is that learners participate in the planning, development and evaluation of projects and activities.
Category	Trainer-Driven models

Template	Socratic questioning
Template file	Socratic questioning_UoL.zip
Title of UoL example	Traditional Spanish Festivals
Use of IMS LD	Level A, Discussion/Forum
Use of IMS QTI (which question types?)	No
Educational technique used (template)	Socratic Questioning
Description of the UoL	Unit of Learning for Basic learners of Spanish (A2). The use of conditional sentences is one of the prerequisites to acquire a basic level of spanish.
Authors	Patricia Santos & Javier Melero
References	Content adapted from: http://ec.europa.eu/spain/ele_es.htm#A2
Description	In the Socratic approach to teaching, the trainer poses thoughtful questions to help learners learn.
Category	Trainer-driven models

Template	One-minute-paper
Template file	One-minute-paper_UoL.zip
Title of UoL	Preterito Indefinido
Use of IMS LD	Level A, Discussion/Forum, Chat
Use of IMS QTI	None
Educational technique	One-paper-minute
Description of the UoL	Unit of Learning for Basic learners of Spanish (A2) teaching the use of "Preterito Indefinido" tense
Authors	Patricia Santos & Javier Melero
References	Content adapted from: http://ec.europa.eu/spain/ele_es.htm#A2
Description	Learners quickly write thoughts to make level of understanding explicit. Feedback to trainer.
Category	Trainer-driven models

Template	Summative Assessment
Template file	Summative-Assessment_UoL-TEMPLATE.zip
Title of UoL	Proverbs and Sayings
Use of IMS LD	Level B (test)
Use of IMS QTI	Yes, fill in the blank and multiple choice questions
Educational technique	Summative Assessment
Description of the UoL	Unit of Learning for Advanced learners of Spanish (C1). This UoLs aims to learn most common Spanish proverbs and sayings to improve speaking fluency, social interaction and understanding.
Authors	Patricia Santos & Javier Melero
References	Content adapted from: http://www.lacunza.com/profesores/rinconestudiante/avanzado.php http://ec.europa.eu/spain/elementos.htm#C1
Description	UoL for making the students memorize concepts/ideas/rules...
Category	Assessment-model driven

Template	Formative assessment
Template file	Formative-Assessment_UoL-TEMPLATE.zip
Title of UoL	Uses of "Se"
Use of IMS LD	Level B
Use of IMS QTI	Yes, Fill in the Blank and multiple choice questions
Educational technique	Formative Assessment
Description of the UoL	Unit of Learning for Advanced learners of Spanish (C1). Uses of "Se" tenses are one of the prerequisites to acquire an advanced level of Spanish.
Authors	Patricia Santos & Javier Melero
References	Content adapted from: http://www.123teachme.com/learn_spanish/exercises_quizzes_1
Description	This UoL includes a set of questionnaires and activities to reinforce concepts.
Category	Assessment-oriented models

Template	Adaptive UoL
Template file	Adaptive_UoL-TEMPLATE.zip
Title of UoL	Adaptive UoL: Verbs&Vocabulary
Use of IMS LD	Level B
Use of IMS QTI	No
Educational technique	Adaptive UoL
Description of the UoL	Unit of Learning for begginers learners of Spanish (A1). Adaptive UoL that offers two different main topics, Verbs and Vocabulary, and adapt the content depending on the responses of the student.
Authors	Mar Pérez Sanagustín
References	None
Description	This adaptive UoL allows to desing a UoL in which the questions change according to the answers given by the students.
Category	Adaptive-driven models

Template	Jigsaw
Template file	Jigsaw_UoL-TEMPLATE.zip
Title of UoL example	History and work of a impressive playwright (Jigsaw)
Use of IMS LD	Level A, use chat and forum widgets
Use of IMS QTI	No
Educational technique	Jigsaw
Description of the UoL	Unit of Learning for advanced learners of Spanish (C1). It aims at revising some advanced verbal tenses in Spanish as well as the reading comprehension. It encourages learners to summarize information reconstructing arguments and producing structured texts on complex subjects using their own words.
Authors	Davinia Hernández-Leo
References	Template available in Collage authoring tool. Content adapted from http://cvu.rediris.es/pub/bscw.cgi/d860522/UN%20DRAMATURGO%20GENIAL,%20ANTONIO%20BUERO%20VALLEJO.pdf (by Carolina Castro)
Description	Jigsaw CLFP organizes a complex learning flow for a context in which several small groups are facing the study of a lot of information for the resolution of the same problem [7]. The activity flow is structured in three phases: i) a first phase in which an individual or initial group studies a particular subproblem, ii) a second phase in which the students that are involved in the same problem are grouped= in Expert groups for exchanging ideas, and iii) a third phase in which the students are grouped in Jigsaw groups formed by one expert in each subproblem to solve the whole problem.
Category	Trainer-driven models

Template	Think-Pair-Share-COLLAGE
Template file	Think-Pair-Share_UoL-TEMPLATE-COLLAGE.zip
Title of UoL example	Singing in subjunctive (Think-Pair-Share)
Use of IMS LD	Level A, use chat and forum widgets
Use of IMS QTI	None
Educational technique	Think-Pair-Share
Description of the UoL	Unit of Learning for advanced learners of Spanish (C1-C2). It aims at reinforcing the practice of understanding Spanish and recognizing the subjunctive verbal tenses and their meaning. Includes listening to a song.
Authors	Davinia Hernández-Leo
References	Template derived from Prolix and Collage collections of pedagogic models Content adapted at http://cvu.rediris.es/pub/bscw.cgi/d639748/CANTANDO%20EN%20SUBJUNTIVO.pdf (by Agustín Yagüe)
Description	Break long presentations: learners think individually, discuss in pairs, then report back to trainer.
Category	Trainer-driven models

Template	360°
Template file	http://docs.google.com/Doc?docid=0AZKdQUn8LJDBZGNiYzJqOThfNzNnOGhkc2Jjcw&hl=en
Title of UoL example	Spanish Composition "La última novela" - 360-assessment
Use of IMS LD	Level B
Use of IMS QTI	None
Educational technique	360 degree feedback assessment
Description of the UOL	360 degree feedback UoL. The teacher creates groups of 3 students.= One student has to upload a document to be assessed. The other two students have to assess the document. The main student will receive the feedback and s/he can answer to this feedback. Once the student has the feedback s/he can use it to improve his/her document. The student do a second submission for the teacher. The teacher assess the document. S/he can see the feedback sent by all the students. Finally the student can check his/her feedback and his / her final mark. To create the different groups of students, the teacher has to publish the UoL and create one "run" for each group of students. The teacher has to be added in each run created.
Authors	FBM-UPF
References	Content adapted from: http://cvc.cervantes.es/aula/lecturas/avanzado/lectura_12/exto/
Description	<i>Description 360-degree-feedback-Spanish_composition.zip</i> : assessment of tasks, students evaluate tasks of their class-mates, think individually, discuss in group, then report back to partner, self-assessment.
Category	discussion-oriented model, collaborative-assessment-driven model

Template	Peer assessment
Title of UoL example	Peer assessment
Use of IMS LD	level B
Use of IMS QTI	none
Educational technique	peer-assessment
Description of the UoL	<p>Peer-assessment feedback UoL. The teacher creates groups of 2 students. One student has to upload a document to be assessed. The other has to assess the document. The main student will receive the feedback and s/he can answer to this feedback. Once the student has the feedback s/he can use it to improve his/her document. The student do a second submission for the teacher. The teacher assess the document.S/he can see the feedback sent by all the students. Finally the student can check his/her feedback and his / her final mark.</p> <p>For creating the different groups of two students, the teacher have to publish the UoL and create one "run" for each group of students. The teacher has to be added in each run created.</p>
Authors	FBM-UPF
References	Content adapted at: http://cvc.cervantes.es/aula/lecturas/avanzado/lectura_12/texto/
Description	<i>Description peer-assessment-Spanish_composition.zip</i> : assessment of tasks,evaluation in pairs, self-assessment.
Category	discussion-oriented model, collaborative-assessment-driven model

Template	Learning Outcomes negotiation UOL
Template file	Learning_Outcomes_negotiation_UoL-TEMPLATE.zip
Title of UoL example	Learning Outcomes negotiation UOL.zip
Use of IMS LD	Level B
Use of IMS QTI	None
Educational technique used	Inquiry based learning
Description of the UoL	The UOL leads learners through the process of defining their learning outcomes, with automated monitoring provided for the teacher.
Authors	Patricia Santos, Mar Pérez-Sanagustín, Stephen Powell, Paul Sharples, David Griffiths=
References	Nonr
Description	UoL for supporting the trainer in following the students progress by storing and commenting the outcomes from a negotiation process.
Category	Discussion oriented model

Template	Adaptive learning using assessment with properties
Template file	C1- Estilo Indirecto y Lenguaje Coloquial - Adaptive assessment.zip
Title of UoL example	Estilo Indirecto y Lenguaje Coloquial
Use of IMS LD	Level B
Use of IMS QTI	Yes: multiple choice, yes/no and fill in the blank question
Educational technique	Adaptive UoL
Description of the UoL	Unit of Learning for beginners learners of Spanish (A1). Adaptive UoL that offers two different main topics, Verbs and Vocabulary, and adapt the content depending on the responses of the student.
Authors	Patricia Santos & Javier Melero
Reference	Content adapted from: http://www.materialesdelengua.org/LENGUA/comunicacion/registros/lenguacoloquial.htm http://www.escuelai.com/gramatica/estilo-indir.html http://www.escuelai.com/gramatica/estilo-directo.html
Description	Adaptive UoL that offers two different main topics, Verbs and Vocabulary, and adapt the content depending on the responses of the student.
Category	assessment driven model

8.2 Appendix 2. Astro Learning Design Player Quick Start Guide

8.2.1 Prerequisites

You have read the “Installing CopperCore (CCRT) environment” guide and understand how to upload Learning Designs to CopperCore. Additionally, you will be familiar with using the linktool and using either the OUNL or SLeD player.

8.2.2 Introduction

The Astro LD Player is an alternative runtime environment to using either the OUNL web player or SLeD. At this present time it is provided as an optional alternative to the default LD player (SLeD). This means you will have to manually navigate to the Astro web address. Astro’s design philosophy is to try and break from the traditional “tree view” outline which previous LD players have been based. Instead, Astro also seeks to provide an additional method of navigation, known as the “filmstrip”. Using this navigation method, activities are presented in a linear fashion, in an attempt to hide some of the Learning Design structure, namely

Unit of Learning -> Play -> Act -> RolePart -> Activity Structure/Learning Activity/Support Activity

Instead, a user simply sees a set of linear activities which appear inside a clickable strip.



Clicking one of these activities within the filmstrip results in the content associated with it being opened up in smaller windows. These windows are designed to be able to be maximized, minimized and draggable. This is a similar ethos to how the widget services were offered within the SLeD player. Note that there still remains a “tree view” in Astro. The tree view is slightly different to previous LD players in that it also shows the whole of the Learning Design structure and does not seek to hide completely, future events. Events that are not yet accessible in the Learning Design are marked as “locked”. This is a design feature and is intended to give the user a feel of where he/she is in the Learning Design. A common complaint with hiding future events completely is that users feel they don’t have a sense of where they are in the Learning Design and what might be coming next.

8.2.3 What do you need

The latest download (v2.0.0 beta) contains all of the required applications to use Astro. To view the Astro player, you can use the latest versions of most of the commonly used browsers. This includes Microsoft Internet Explorer, Mozilla Firefox and Safari and Google Chrome. You will also need an active internet connection, as Astro also relies on several other libraries that are found on the web.

8.2.4 Prerequisites before using Astro

Read the “Installing CopperCore (CCRT) environment”, particularly points 1-12 of the “Getting started using the Link Tool and Sled” section, which describes how to upload Learning Designs and create runs.

8.2.5 Getting started using Astro

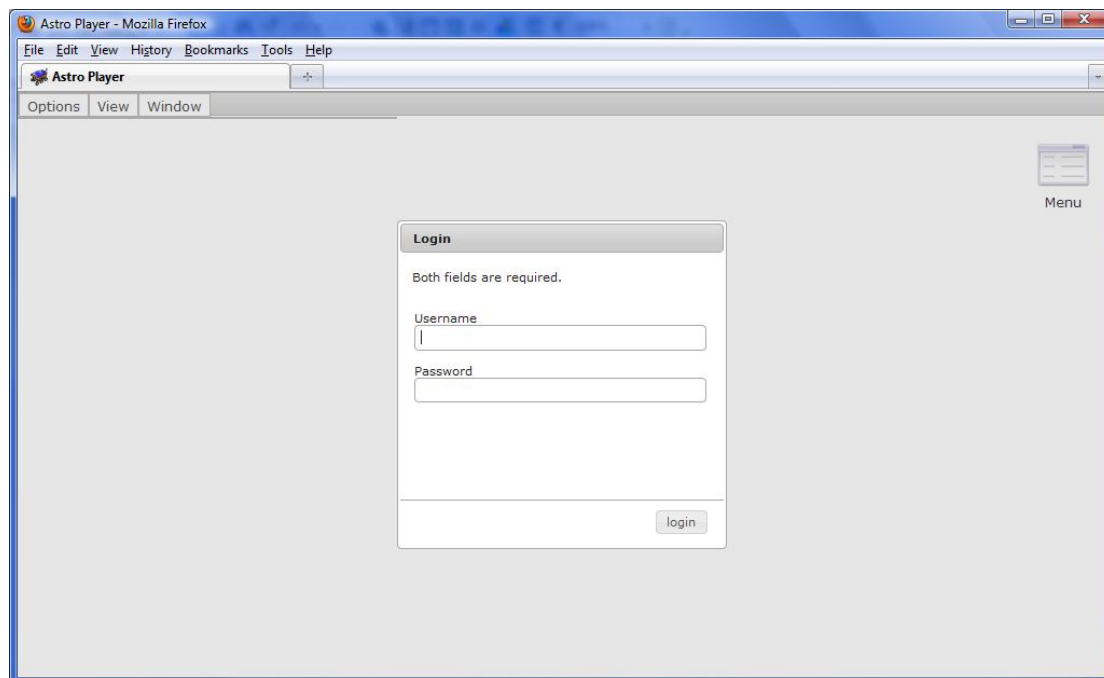
Once you have uploaded a Learning Design and created a run as previously described, you will be able to use Astro to access it.

1) Point your browser to...

<http://localhost:8080/astro>

...(or your own server address if you have changed this elsewhere)

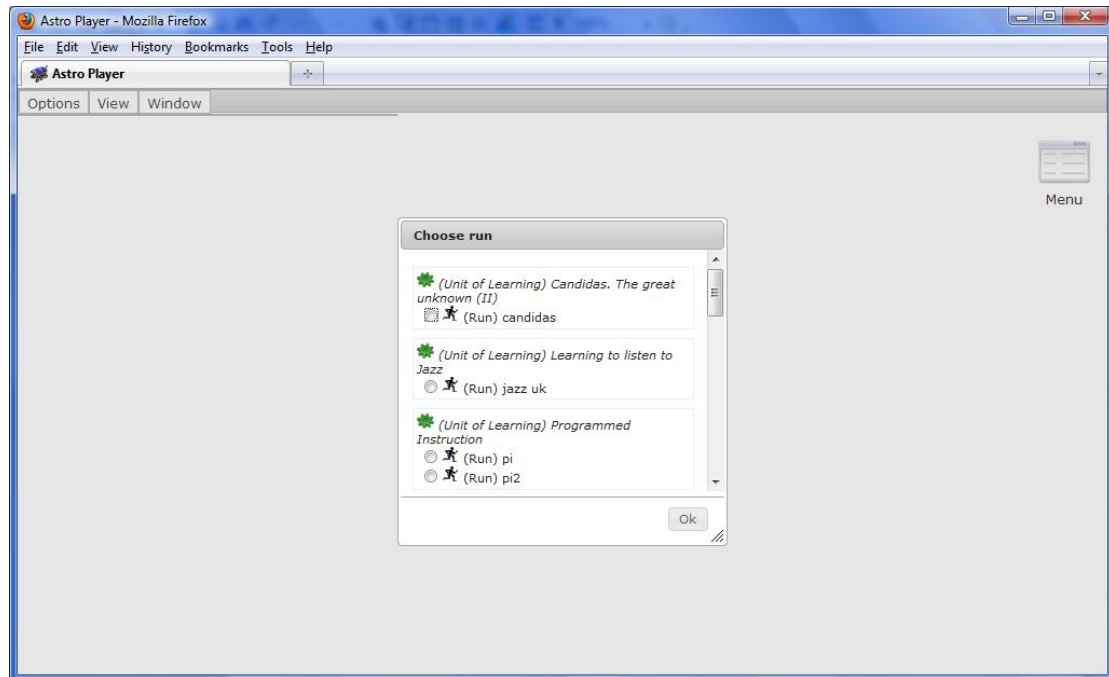
You should see the Astro login screen.



Enter your username and password here then click “login”.

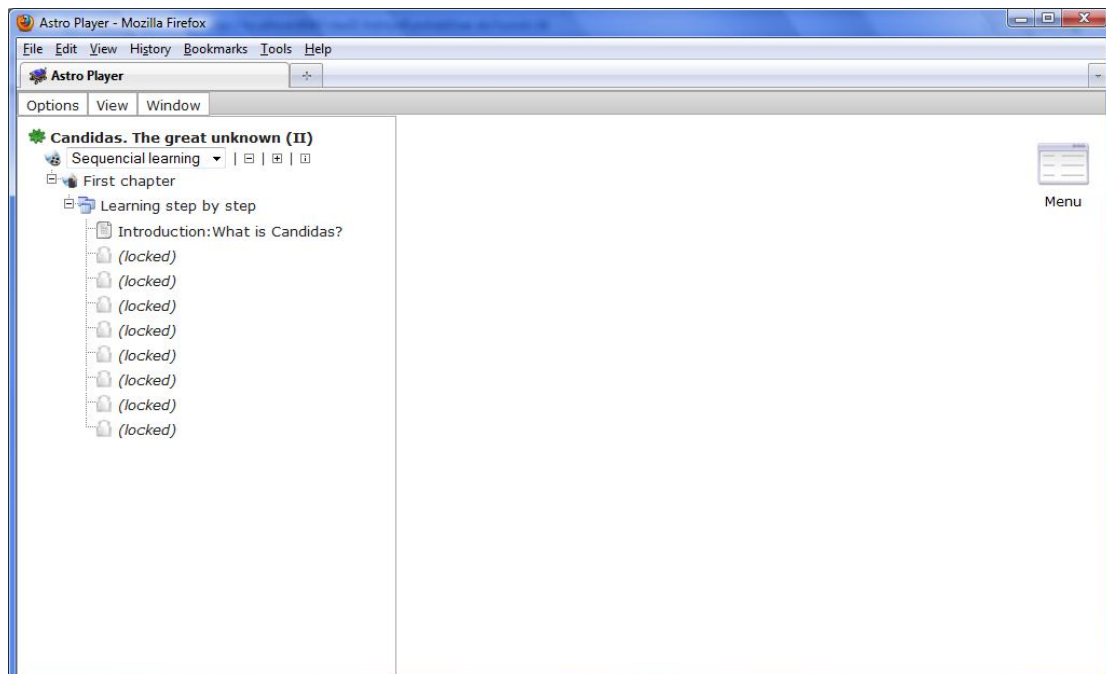
Next you should see a dialog box which asks you to choose which run you want load.

2) Next you should see a dialog box which asks you to choose which run you want load...



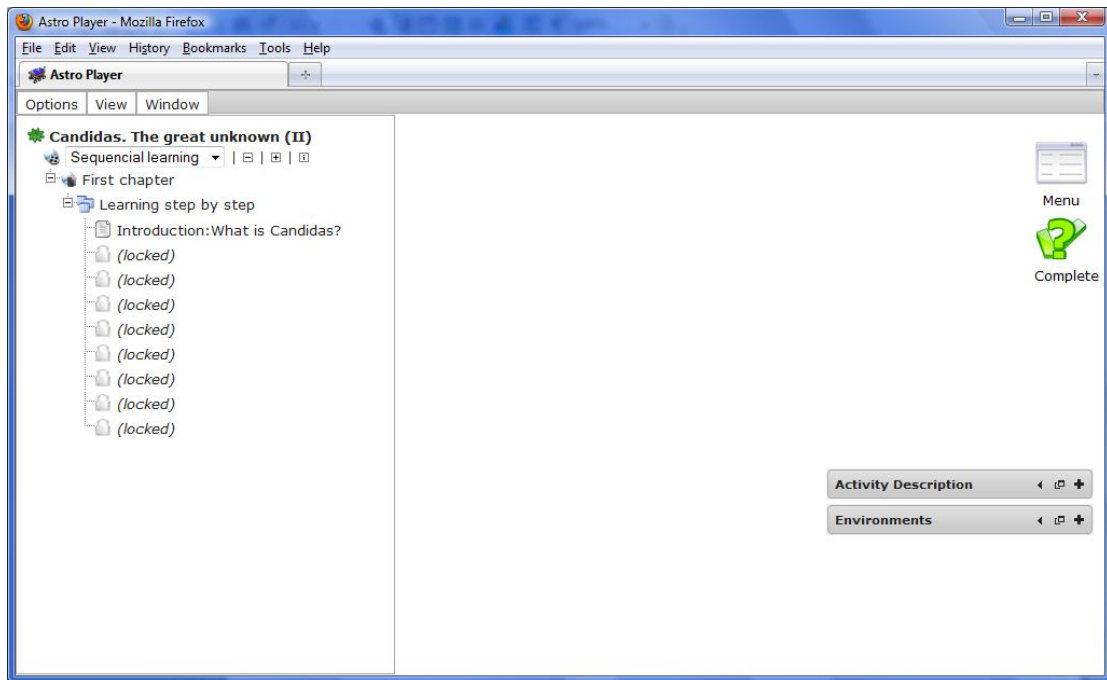
Choose one of the runs (which appear under each Unit of Learning) as a radio button and click "ok"

3) You should now see the selected run appear in Astro...

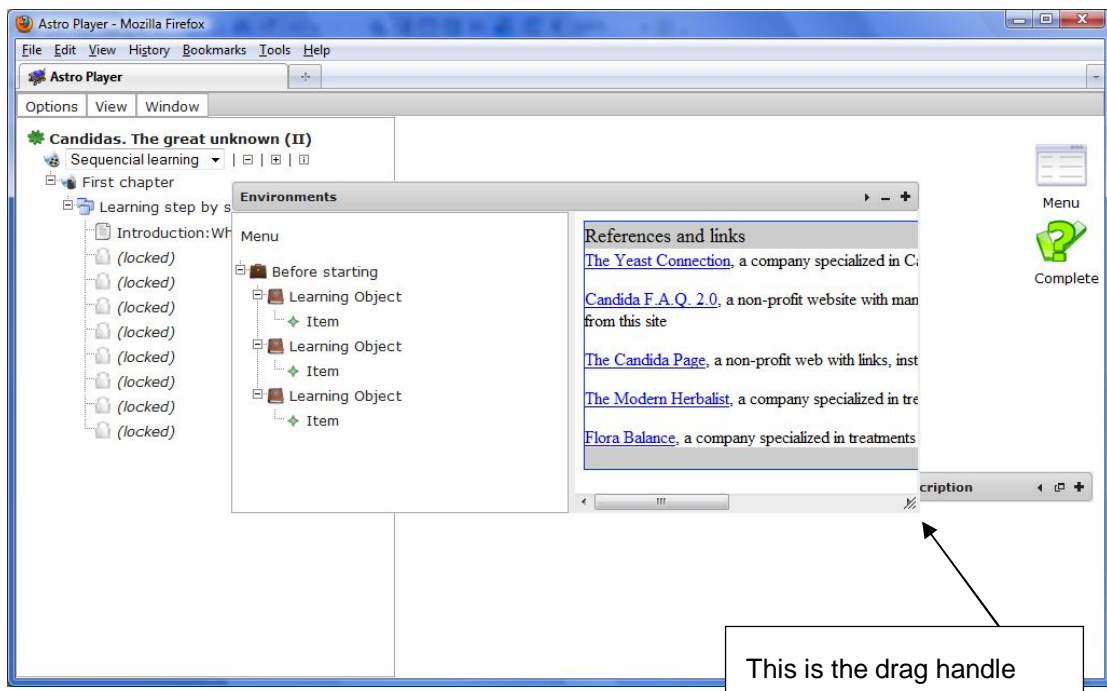


At first glance, Astro may not look all that much dissimilar to SLeD. By default it has opened with a Tree view structure running down the left-hand side. However, if you click on one of the branches of the tree, you will see that one or more windows will open and then minimize to the bottom right of the screen.

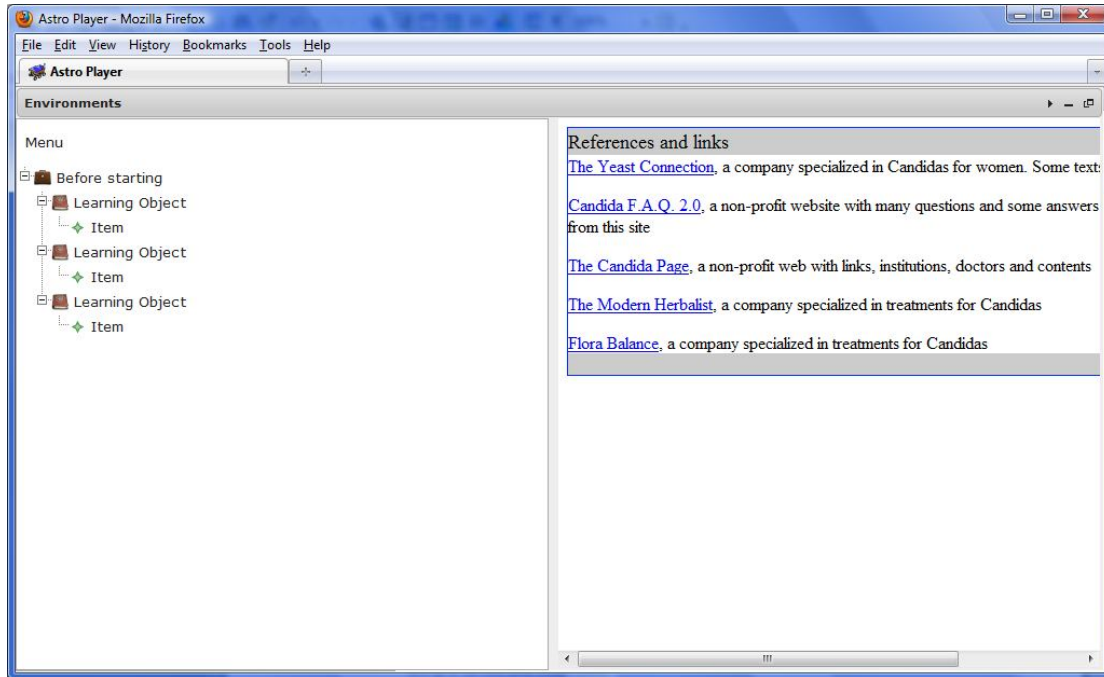
D6.4 – Compilation of internal deliverable outcomes ID6.7 – ID6.1



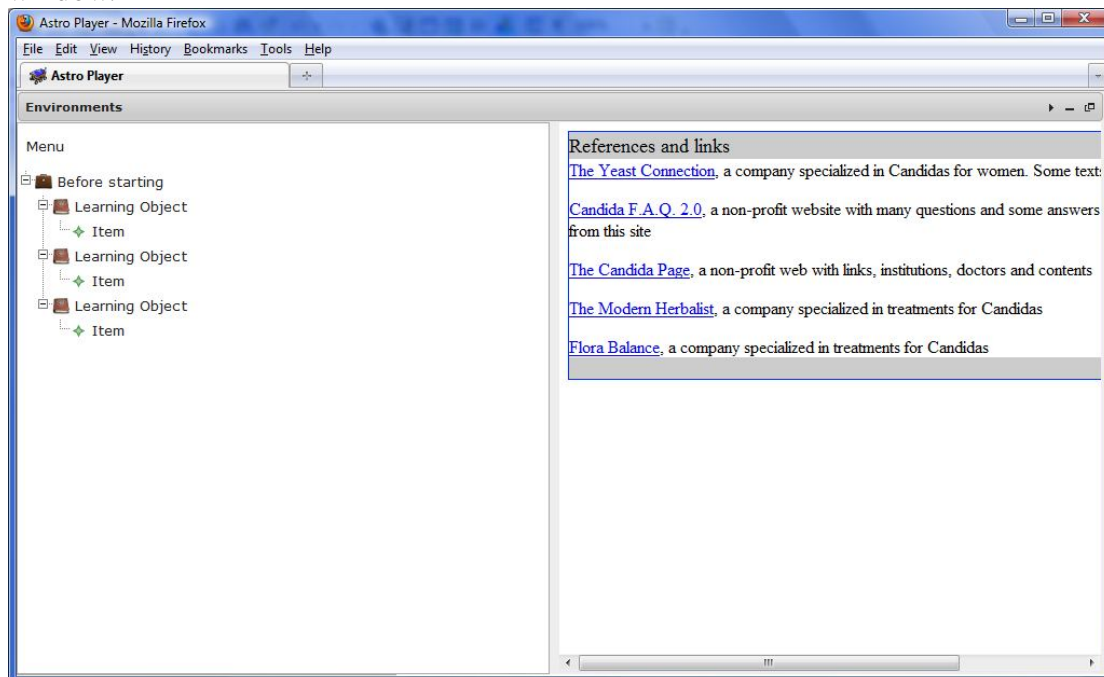
If you click on the second icon along on one of these minimized windows, it will open to a specific size. Try to grab the bottom right of this window to make it large as indicated in the following figure.



You can also maximize this window, so that it appears full screen. Click the third icon along to do this (the “+” icon).



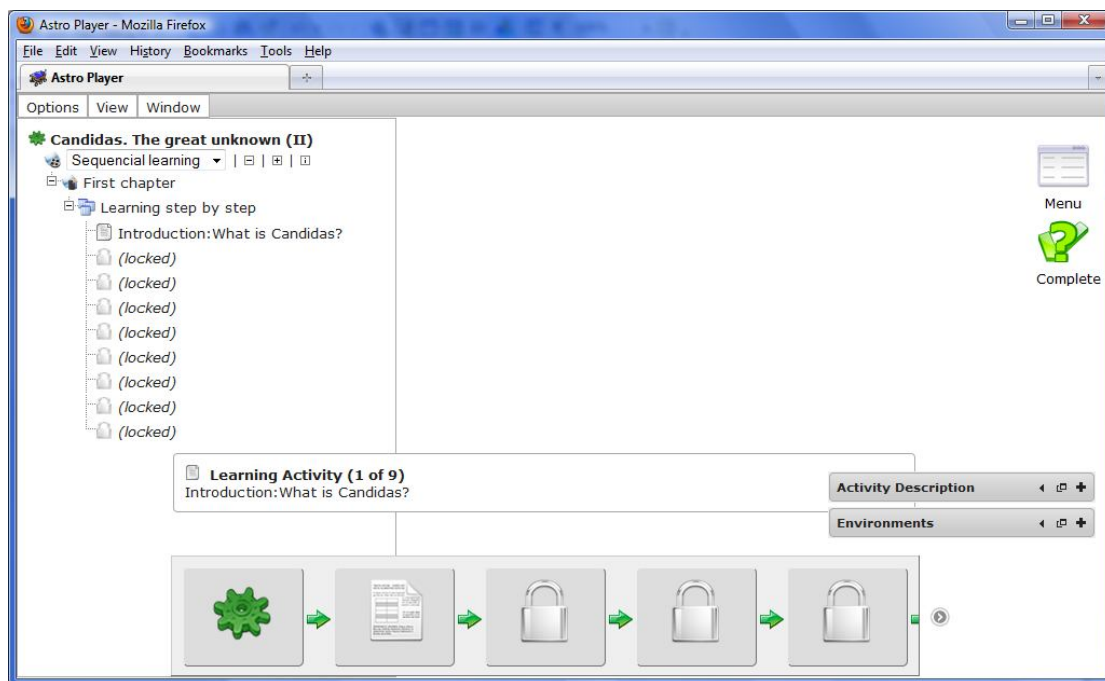
Clicking the first icon on the window, toggles the menu view for this particular window.



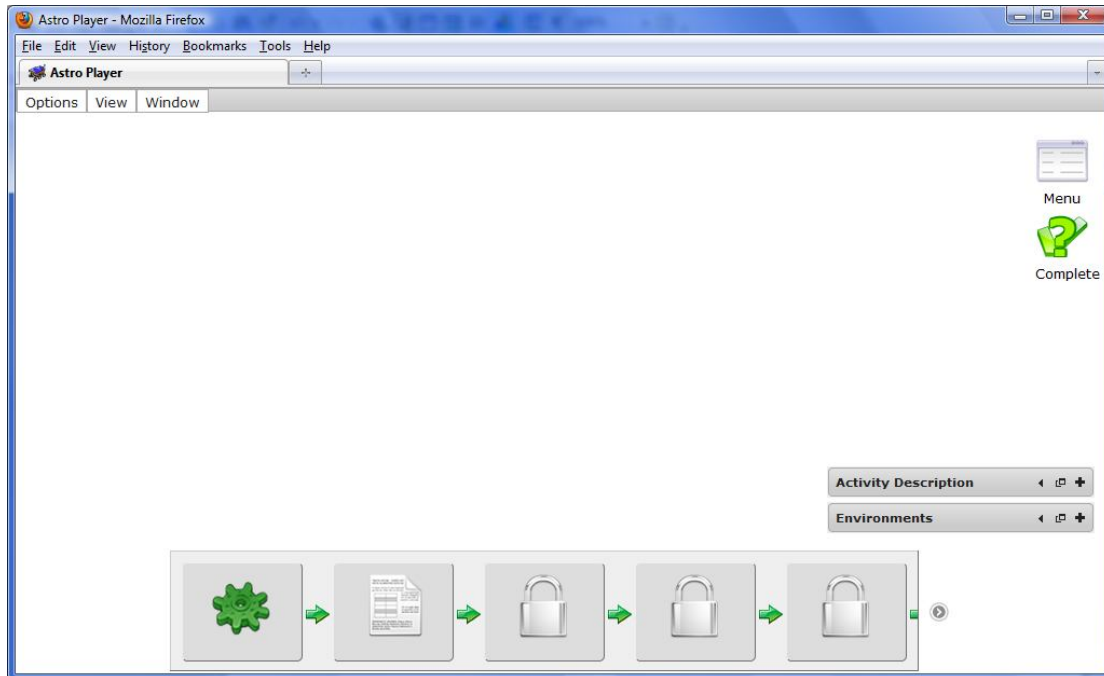
Now you are able to make use of all of the screen to view the content. Click the minimize icon to reduce the size of this window. Note – you can also use the menu option “window -> minimize all” to go back to the original window state.

4) To use the Filmstrip navigation click the menu option “view -> toggle filmstrip”. You should now see both the Tree view and the filmstrip together...

Hover over an item in the Filmstrip to see a description of what it contains and where it appears in the Learning Design. Clicking an item has the same effect as clicking one of the leaf nodes in the Tree View. Notice how the Filmstrip does not show anything other than “activities”. The first activity in the filmstrip is always the overview, which is the only exception to this rule. This equates to the highest level information for the Learning Design, found at the UOL level. (In fact it has the same behaviour as clicking the top node in the Tree view.) Any acts or activity-structures for example, are hidden in this view, unlike the tree view.



5) Next click the menu option “view -> toggle treeview”. The treeview should now be hidden (you can get it back later by clicking the menu option “view -> toggle treeview” again)



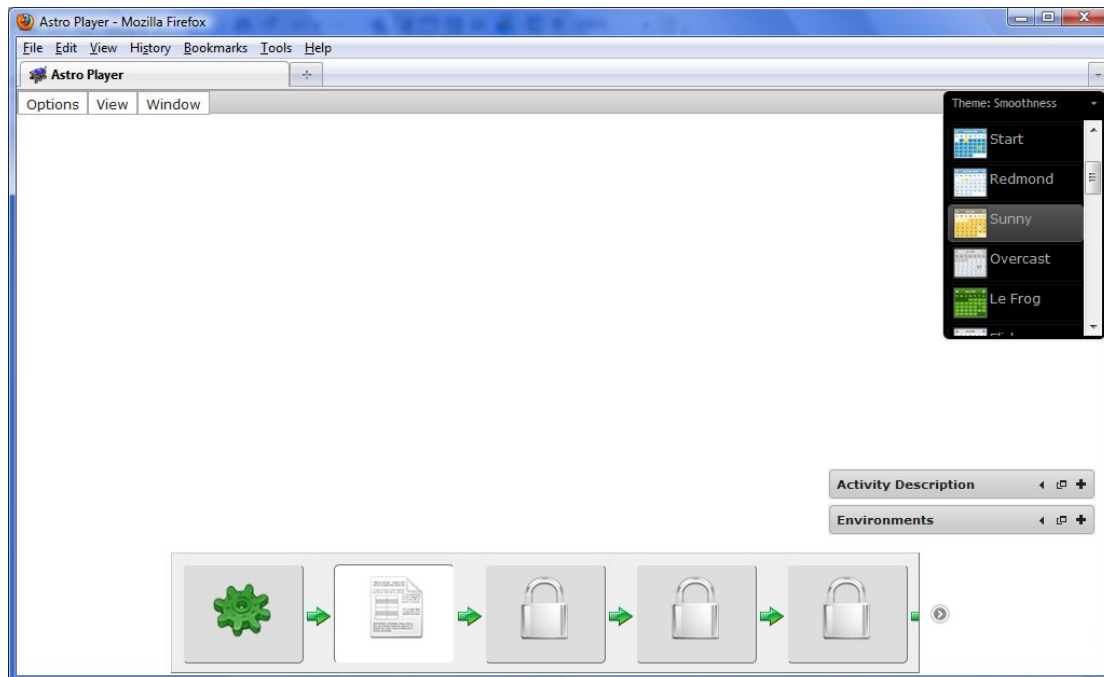
6) To select an activity in the filmstrip, simply click the icon, which will represent an activity. If the activity is not available yet, then it will be shown as a lock – meaning that the CopperCore state engine has determined it to be not available yet. (this could be because it is in a future act or is waiting for a property to be set, for example.) This behaviour is the same in the treeview also and is different to how SLeD displays the Learning Design Structure.

7) You can use your mouse to scroll forwards or backward in the filmstrip. There are also arrows either side of the filmstrip to navigate backwards and forwards. (However, this depends on how many total activities there are)

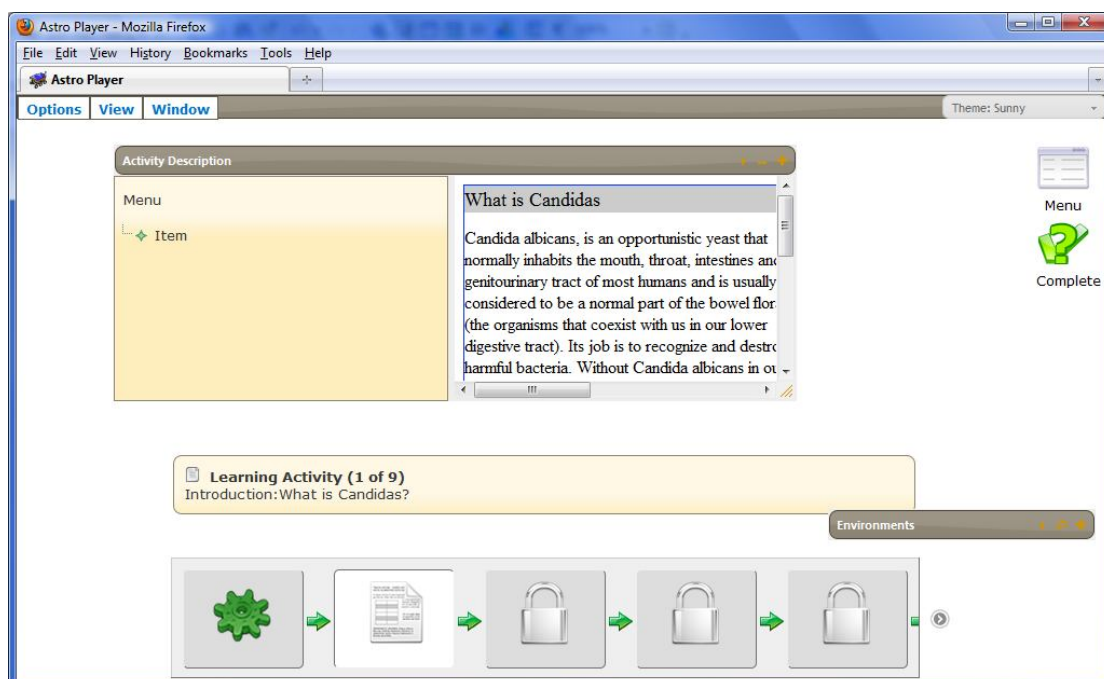
8) If an activity is set to “user-choice”, which means you have to explicitly click something to say you have done it, then a green icon will appear near the top right of the screen with the word “complete” under it. Click this to set it as completed. The player should update automatically and reflect any changes as a result of this action.

9).Use the menu option “options -> change run”, to select a different run.

10) To show a list of styles you can use with Astro se the menu option “view -> toggle themes widget”. The list will appear at the uppermost top right of the screen.



Click the widget and then select one of the themes from the drop down list which appears. This will have the effect of changing the look and feel of Astro.



8.3 Appendix 3: Wookie Developers Guide

For use with Wookie Engine v0.9

8.3.1 Introduction

8.3.1.1 How widgets work

A widget is a miniature web application designed to work within a particular kind of framework, called a ***widget engine***. Unlike normal web applications, a widget doesn't execute server-side code directly (e.g. written in PHP), but instead executes code in the browser using JavaScript, and access a range of web services offered by the widget engine.

Wookie Engine is a widget engine that is designed to enable widgets to be used in a wide range of web applications; we call a web application that has widgets in it a ***widget container*** (or just plain *container*).

Widget containers take care of all the main business of being a web application, like letting users log in, managing content and so on – or whatever it is they do – while Wookie Engine just supplies the functionality to add widgets to the mix.

When you create a widget, you are only interacting directly with the widget engine, and not the container – the container may sometimes set preferences your widget may use (like the user's name to display in your widget), but for the most part you are calling the services offered by the widget engine.

So when you develop a widget for Wookie, you are enabling it to be presented in a range of web applications, which may include things like Wordpress, Elgg, Moodle, Sharepoint and so on.

Wookie enables widgets to be used in these applications through the use of ***plugins***. This means there is some code which is native to that web application's framework that can talk to Wookie and request widgets.

This guide focuses on what you need to know in order to create *widgets* that you can upload to the Wookie *widget engine* that can then be made available to *widget containers* through their *plugins*.

There are two companion guides to this document you may want to take a look at:

- 1 The ***Wookie Server Administrators Guide*** contains information about how to install, configure and run the *widget engine* itself
7. The ***Wookie Plugin Developer's Guide*** contains information about how to write a *plugin* for a web application so it can become a *widget container*

8.3.2 Creating Basic Widgets

8.3.2.1 Planning your widget

A good widget focuses on performing a single task well with a very clear and appropriate graphical style. A good way to get started with designing a widget is to create a graphical mockup of the widget in its various states.

For this tutorial, we're creating a rather silly weather widget that just shows today's forecast. Its really only of use to people in offices without windows, but it should demonstrate the basics of creating and deploying a widget.

Our weather widget has a very simple layout, with a main display and a settings display where you can choose your city:

8.3.2.2 Widget file structure

A widget typically consists of the following files:

- The widget manifest file (config.xml), which describes the widget
- A HTML start page, typically "index.html"
- One or more Javascript (.js) files that implement widget's functionality
- One or more stylesheets (.css) that control the appearance of the widget
- An icon (icon.png) for the widget
- Additional media assets such as images.

For example, an unpackaged weather widget might look like this:

	config.xml
	index.html
	weather.js
	weather.css
	icon.png
images	
	cloudy.png
	rainy.png
	sunny.png
	snowy.png

Widgets can also support internationalisation by organising these files into *localised folders*. Each localised folder can contain files that override the defaults when the widget is deployed in a particular location. For example, we can offer the widget with French and German localised resources:

	config.xml
	index.html
	weather.js
	weather.css
	icon.png
images	
	cloudy.png
	rainy.png

	sunny.png	
	snowy.png	
locales		
	de	
		index.html
		weather.css
	fr	
		index.html

Localised folders need to be placed in a top level folder called “locales”, and be named according to the appropriate ISO two-letter language code, and can contain any resources or structures you want. A localised folder can contain just the HTML file, or can override other resources such as the icon, stylesheet, javascript and so on. You must, however, within your localised start file reference the appropriate assets – the widget engine will launch using the appropriate start file in your localised folder, but it won’t rewrite any URLs for you.

If you don’t use localised folders, then the engine will look for the default index.html file, or whichever file is defined as the starting file in the widget manifest (config.xml).

A good starting point for developing a new widget is to take an existing widget as your starting point, and to replace each of the parts with your own code when you’re ready.

8.3.2.3 The start file

Your start file is a HTML document that is loaded by Wookie into the iFrame in the target container, and is the starting point for rendering your widget.

The design of the start file is entirely up to you, but in general it should set out the structural elements of the widget (e.g. named <div> tags for each item of the Widget’s interface), and reference your Javascript and CSS files.

In general the start file can be quite small and simple, as usually widgets make extensive use of Javascript and CSS for dynamic functionality and appearance.

Our weather widget’s index.html file has a <div> for the front and the back (where the settings are); the front <div> has blocks for the content – the weather picture, the city, and temperature. We also have a place for the buttons to go. We also put some links in for our stylesheet, and our javascript.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Weather</title>
```

```

<script type='text/javascript' src='weather.js' charset='utf-
8'></script>

<link rel="stylesheet" href="weather.css" type="text/css"/>
</head>
<body onload="weather()">
  <div id="front">
    <div id="weather"></div>
    <div id="city"></div>
    <div id="temperature"></div>
    <div id="settings">
      <input type="submit" class="groovybutton" id ="settings"
value="settings" onClick="showBack()">
    </div>

  </div>
  <div id="back">
    <div id="controls">
      <select id="city_selector" name="city_selector"
onchange="setCity()">
        <option>Belfast</option>
        <option>Birmingham</option>
        <option>London</option>
        <option selected>Manchester</option>
      </select>
      <input type="submit" class="groovybutton" id ="done"
value="Done" onClick="showFront()">
    </div>
  </div>
</body>

<!-- fix forIE caching - must go here even though its in the wrong place
<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
</HEAD>
-->
</html>

```

8.3.2.4 Writing your Javascript code

The JavaScript you create really depends on what you want your widget to be able to do; the main thing you need to bear in mind are the kinds of services that you can call upon the Wookie engine to do for you. Wookie can:

5. Store and retrieve your user's preferences, or any other settings unique to the widget.

6. Maintain data shared between all instances of your widget in a common context; for example, a chat log and buddy list for a chat widget.
7. Send events between widgets in the same context; for example, if a user sets their location in one widget on their profile page, the widget can send a message to other widgets on the same page.
8. Handle calls to external web services and feeds through a proxy service. This is important, as otherwise your widget cannot communicate with other services as this poses a security risk.

For all of these services, your widget can make use of the shared Widget object that Wookie makes available to your JavaScript code. To invoke the Widget object, you simply call one of its functions; for example:

```
Widget.preferences.setItem("foo", "bar");
```

8.3.2.5 Working with user preferences

As you've probably guessed from reading the previous section, Wookie offers a preferences service for your widget based on key-value pairs. There are two methods:

```
Widget.preferences.getItem(key)
```

```
Widget.preferences.setItem(key, Value)
```

The key can be any arbitrary string, and the value can be any kind of variable; typically a string, some XML, or a JSON object.

Defaults

As well as dynamically setting preferences in JavaScript, you can also set default preferences in your config.xml file, for example:

```
<preference name="city" value="manchester"/>
```

These values are loaded by Wookie when you install a widget, and are set each time someone instantiates the widget. You can, however, override these default values in your scripts. For more information, see the section titled Creating a widget manifest file.

8.3.2.6 Contextual preferences

While for the most part the preferences for a widget are set by your own JavaScript code, you can use the `preferences.getItem(key)` method to gain access to properties that have been set by the container that is displaying your widget. This is because Wookie plugins have access to a service that can set default values for widgets; for more information, see the *Wookie Plugin Developer's Guide*. In general the kinds of defaults your widget can access are things like the user's display name, and whether the user has moderator rights.

While the container can set any properties it wants to, in order to take advantage of these values your widget needs to know what values to ask for. The *Wookie Plugin Developer's Guide* recommends that containers set the following values where appropriate:

- **username**, as a string containing a display name for the user (not necessarily their login name or account)

- **moderator**, as a Boolean value, with True indicating that the current user has moderation privileges within the context the widget is located in.

So, for example, a chat widget can call `preferenceForKey("username", fooHandler)` to obtain a chat handle for the current user, and `preferenceForKey("moderator", barHandler)` to check if the current user gets access to the moderator controls, like locking the forum or whatever moderator features the widget wants to offer.

If you include references to container-set defaults, you should make sure that you include code that handles where these values are not set. For example, a chat widget should be able to either generate a random handle (e.g. "chatter#2232") for anonymous (not logged in) users, or display a message saying that chat is only available when logged in to the application. Likewise if "moderator" is not set, then a widget may assume a value of "False" as the default.

This is particularly important when your widget makes use of container-specific properties, as you have to have a strategy for dealing with situations where the widget is being used with a different container that does not set these values.

Our weather widget wants to be able to make use of preferences to maintain the user's location information so they don't have to keep setting it each time they load the widget. To do this we use the `preferences.getItem()` method:

```
function weather(){
    var pref = Widget.preferences.getItem("city");
    if (pref && pref != "null"){
        city = pref;
        startFetchingWeather();
    }
}
```

With this code we're not going to try and fetch any weather data until we've checked for a city preference. If no preference has been set, we just keep the default value.

If the container sets the "city" value based on the user's profile, then the widget will be automatically configured; otherwise the user will have to set it themselves the first time.

If the user does select a city on the "back" of our widget, we want to store this, so:

```
function setCity(){
    var select = document.getElementById("city_selector");
    city = select.options[select.selectedIndex].value;
    Widget.preferences.setItem("city", city);
    fetchWeatherData();
}
```

8.3.2.7 Working with external web services and APIs

Most widgets – or at least, a lot of the more interesting ones – will make use of external services of one kind or another. Whether its to fetch a feed of interesting items, or just to make use of handy shared services for things like unit conversions. To call an external service your widget needs to implement an AJAX request and handle the response, and to do without breaching the security restrictions of the user's browser.

Luckily, Wookie handles the latter aspect for you by providing a built-in proxy service that lets you call services through the same Wookie server that is serving your widget, avoiding the dreaded Same Origin Policy Violation error. The downside is that you have to make sure that any servers your widget needs to call are listed in Wookie's whitelist.

For information on how to do this, check the Wookie Engine Administrator's Guide; however, you should declare the sites you need to access in your widget configuration file.

To invoke an external service, you need to construct a proxy URL. Wookie provides a handy method for this:

```
Widget.proxify(url)
```

This method returns a "proxified" version of a URL you provide it, for example:

```
Widget.proxify(http://www.foo.bar/xyz);
```

Will return something like:

http://www.mywookieserver.com:8080/proxy?instanceid_key=452462363456453653456&url=http://www.foo.bar/xyz

The actual details of the URL the method generates aren't all that interesting, the main thing is that you can go ahead and call it, assuming you've added "www.foo.bar" to Wookie's whitelist.

There are many patterns for making AJAX service calls; you can use existing helper libraries such as Prototype, or just code it directly yourself. As long as you proxify your URL, you should be fine.

If your service requires authentication, however, you need to check the section in the Advanced chapter.

For our weather widget, we're going to load an RSS feed from the BBC's Weather service for our user's home city. We implement this with a series of functions; first we have a scheduler that keeps checking the feed every so often:

```
function startFetchingWeather()
{
    fetchWeatherData();
    timer = setInterval ('fetchWeatherData();', 240000);
}
```

Next, we have the function that actually fetches the feed, based on a lookup of the user's city:

```
var cities = new Array;
cities['belfast'] = '1';
cities['birmingham'] = '2';
cities['manchester'] = '9';
cities['london'] = '8';

function fetchWeatherData(){
    var loc =
"http://feeds.bbc.co.uk/weather/feeds/rss/5day/world/"+cities[city]+".xml";
    loc = Widget.proxify(loc);
```

```

var xml_request = new XMLHttpRequest();
xml_request.open("GET", loc, true);
xml_request.onreadystatechange = function()
{
    if(xml_request.readyState == 4 && xml_request.status == 200){
        parseXMLData(xml_request.responseXML);
        buildOutput();
    }
}
xml_request.setRequestHeader("Cache-Control", "no-cache");
xml_request.send(null);
}

```

This function calls the service, and then on successful completion invokes our methods for parsing the feed and updating the display. The parse function is rather rudimentary:

```

var iconTable = new Array;
iconTable['sunny'] = 'images/sunny.png';
iconTable['sunnyintervals'] = 'images/sunny.png';
iconTable['rainy'] = 'images/rainy.png';
iconTable['lightshowers'] = 'images/rainy.png';
iconTable['drizzle'] = 'images/rainy.png';
iconTable['lightrain'] = 'images/rainy.png';
iconTable['heavyrain'] = 'images/rainy.png';
iconTable['snowy'] = 'images/snowy.png';
iconTable['cloudy'] = 'images/cloudy.png';

function parseXMLData(xmlobject){
    // The BBC formats weather feeds like this:
    // <title>Wednesday: cloudy, Max Temp: 10&#xB0;C (50&#xB0;F), Min Temp:
    10&#xB0;C (50&#xB0;F)</title>

    var root = xmlobject.getElementsByTagName('rss')[0];
    var channel = root.getElementsByTagName('channel')[0];
    var item = channel.getElementsByTagName("item")[0];
    var title = item.getElementsByTagName("title")[0].firstChild.nodeValue;

    var desc = title.split(":")[1];
    desc = desc.split(",")[0];
    desc = desc.split(" ").join(""); //remove whitespace

    var temperature = title.split(":")[2];
    temperature = temperature.split(",")[0];
    celsius = temperature.split("(")[0];
    fahrenheit = temperature.split("(")[1];
    icon = '';
}

```

```
}
```

This sets the values of the properties we then use to update the display:

```
function updateDisplay()
{
    var iconDiv = document.getElementById("weather");
    iconDiv.innerHTML = icon;
    var locationDiv = document.getElementById("city");
    locationDiv.innerHTML = city;
    var tempDiv = document.getElementById("temperature");
    tempDiv.innerHTML = "<p>"+celsius+"</p>";
}
```

Of course you wouldn't really make something this basic for a "real" weather widget – we have nothing here for handling errors, for example - but hopefully you can see the basic methodology for calling a service.

8.3.2.8 Creating a widget manifest file

Before you can deploy a widget on your Wookie server you need to give it a manifest file that describes the widget and provides some configuration details that the engine can use when it renders your widget in a container application.

The manifest file must be called *config.xml*, must be located at the root of your widget's file structure, and must conform to the W3C Widgets: Packaging and Configuration v1.0 specification. You can also provide different language versions of your manifest in localized folders, as described earlier in this guide.

The key things to consider for your manifest are:

- It must have a root element called `<widget>` with attributes for the height and width of the widget. This is important as many containers will display your widget in an iFrame based on this information. It also needs a unique id; a URL is a good way to manage this.
- It must have a `<name>`, and preferably also a `<description>`.
- If you want to provide an icon for your widget, make sure you include an `<icon>` element with a `src` attribute set to the filename of your widget's icon.
- Include a `<content>` element with the `src` attribute set to the filename of your start file; this will default to `index.html`, but it doesn't hurt to make this explicit.
- If your widget uses external services, include an access element for each: `<access uri="http://feeds.bbc.co.uk/weather/" />`
- If you want to set default values for any preferences, include `<preference>` elements with `<name>` and `<value>` attributes.
- If you like, you can add your name in the `<author>` element, and provide a `<license>` containing your copyright information.

There are many other settings you can make in the widget manifest; for more details take a look at the W3C specification.

Our weather widget's manifest looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
  id="http://www.getwookie.org/widgets/weather"
  version="1.0"
  height="125"
  width="125">
  <name>Weather</name>
  <description>A silly Weather widget</description>
  <icon src="icon.png"/>
  <content src="index.html"/>
  <access uri="http://feeds.bbc.co.uk/weather"/>
  <author>Scott Wilson</author>
  <licence>Example license (based on MIT License)
Copyright (c) 2008 The Foo Bar Corp.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
  </licence>
  <preference name="city" value="Manchester"/>
</widget>
```

8.3.2.9 Packaging and deploying widgets with Wookie

Once you've created a widget, including its manifest, you can simply package your widget and upload it to your server.

To package a widget, make a Zip archive of all the files that make up the widget. Make sure that when you do this that the files are at the root of the archive, and not all nested inside a subfolder.

Once you've zipped your widget, change the file extension to ".wgt", and you can upload it to Wookie using the management interface. When you've uploaded your widget you need to allocate it a service type, and add any services required to the proxy whitelist.

For more details, see the *Wookie Engine Administrator's Guide*.

8.3.3 Creating Advanced Widgets

8.3.3.1 Locking and unlocking widgets

In some cases you want to be able to let users lock a widget so that other user's cannot make any changes. To implement this feature you need to be making use of event notifications, so your widget manifest must include:

```
<feature name="comet" required="true"/>
```

To lock or unlock a widget, the API provides two methods:

```
Widget.lock()
```

```
Widget.unlock()
```

The methods can have a callback function attached, but in general this method is fairly safe to call without it.

To make use of locking and unlocking, your widget needs to be able to respond to locking and unlocking events that are pushed from the Wookie server. To do this you need to create handler functions that respond to the events; for example:

```
Widget.onLocked = handleLocked;
```

```
Widget.onUnlocked = handleUnlocked;
```

These declarations assign “handleLocked” to be invoked when the widget receives an event notifying it that the widget has been locked, and “handleUnlocked” to be invoked on receiving an event notifying it that the widget has been unlocked.

For a detailed example of locking, look at the source code of the default chat widget included with Wookie server.

8.3.3.2 Working with shared data

If you want widgets to have collaborative or social functionality, you need to make use of the shared data API. Shared Data provides methods for storing and accessing data that can be shared among all instances of a widget that share a common context.

The meaning of “context” varies according to how the Wookie Plugin is configured to work with the container, but typically means a widget occupying a single space in the system. For example, for Wordpress the context is the blog; for Moodle it’s the course; for Elgg it’s the user profile page. For more information on contexts, see the *Plugin Developer’s Guide*.

To implement this feature you need to be making use of event notifications, so your widget manifest must include:

```
<feature name="comet" required="true"/>
```

The Shared Data API contains the following methods:

```
Widget.appendSharedDataForKey(key, value, callback)
```

```
Widget.setSharedDataForKey(key, value, callback)
```

```
Widget.sharedDataForKey(key, callback)
```

The `appendSharedDataForKey()` method is a safe way of setting a shared value as it will only append the value you set to the end of the existing value of the shared data entry. You can use this method to do things like append messages to a shared chat log, for example.

The calls are asynchronous, and so it is not reliable to just invoke some of the methods on `Widget` without providing a callback handler. For example:

```
Widget.sharedDataForKey("foo", "bar", myHandler);
```

This means that when the service returns a value, your handler is called. If you need to call a series of services, a common way to do this is to chain the handlers together, e.g.:

```
Widget.setSharedDataForKey("foo", myHandler);
function myHandler(data){
    sharedDataForKey("bar", myOtherHandler);
}
function myOtherHandler(data){
    // do something
}
```

Chaining service calls like this prevents your widget from getting into a tangle of colliding responses that can affect the validity of its state.

The get and set methods work in a similar fashion to the preference methods; however, you need to take care of how you use them as multiple users are using your widget, which makes it highly likely that messages will overlap each other causing the state to be inconsistent. In general you should only use `setSharedDataForKey()` and `sharedDataForKey()` for values that are designed to be only ever *set by a single widget instance*, but are designed to be *read by multiple instances*.

For example, if you have a voting widget, you may want to store each user's vote separately, but add them together to make the final score; this makes it possible for the user to change their vote without potentially messing up the tally if several users were to try to change it at the same time.

To generate an instance-specific shared data key, you should invoke the `instanceId` attribute:

```
Widget.instanceId
```

This returns a unique identifier for your widget instance.

Whenever a shared data value is set or appended, widgets receive an event notifying them of the changes. Your widget can handle notifications by setting a function as the event handler in your script; for example:

```
Widget.onSharedUpdate = handleSharedUpdate;
```

This sets the *handleSharedUpdate* method to be invoked whenever there is an event notifying the widget that a shared data value has been updated.

Each event contains an array of keys for the values that have been updated; your widget can use these to determine what actions, if any, that it needs to take. For example:

```
function handleUpdate(keys){
    for (String key:keys){
        if( key.equals("chat"){
            sharedDataForKey("chat", refreshChatDisplay());
        }
    }
}
```

There are many different strategies that can be employed for working with shared data; a good starting point is to look at the code for the default widgets provided with the Wookie Engine.

In some cases, the functionality you need for your widget may be more than the shared data API can offer, in which case you should consider connecting to an external service to implement the features remotely.

8.3.3.3 Working with state coupling events

This feature will be added in v1.0 of Wookie Server.

8.3.3.4 Working with secure services

Whenever your widget needs to call a secure service, you face a range of options for how to handle this situation. Each has its pros and cons.

8.3.3.4.1 ASK THE USER TO DIRECTLY PROVIDE THEIR LOGIN INFORMATION TO THE WIDGET, AND THEN ACCESS THE SERVICE PREFIXING THE AUTHENTICATION DETAILS

This has the benefit of simplicity but is extremely poor practice from the viewpoint of security and user education. Not only is the widget able to act on behalf of the user, it means storing credentials in plain text on the Widget Engine, which requires that the person running that server has to make sure its database is safe from snooping.

If you really, really must do this, then you can add the optional username and password parameters onto the proxified URL for the service.

8.3.3.4.2 USE AN ACCESS TICKET

An alternative to using credentials is to generate a ticket from the service that the user can paste into the widget instead of their credentials. The ticket can be specific to the widget and not usable by other agents or applications, enabling the service to track usage.

The advantage of this approach is nothing sensitive is saved in the Widget Engine database, or transmitted insecurely over the network. The disadvantage is that it shifts responsibility to the service for generating and managing tickets.

To use a ticket-based solution requires adding the appropriate ticket parameter to the service call.

8.3.3.4.3 USE OAuth

OAuth is a standard ticket architecture used by a number of services, including Google's applications and services. OAuth provides a mechanism for an application to request access to a service, prompting the user to login and grant a ticket.

OAuth is the most likely future scenario for accessing secure services in a consistent, user-centric way. Unfortunately, the current version of Wookie has not yet implemented it, so you'll have to wait!

8.3.4 Appendix A to Wookie Widget Developers Guide: Widget API Reference

8.3.4.1 Core W3C features

Note that the W3C specification is evolving, and these features are subject to change.

```
interface Widget {
  readonly attribute DOMString viewMode;
  readonly attribute DOMString locale;
  readonly attribute DOMString identifier;
  readonly attribute DOMString authorInfo;
  readonly attribute DOMString authorEmail;
  readonly attribute DOMString authorHref;
  readonly attribute DOMString name;
  readonly attribute DOMString description;
  readonly attribute DOMString version;
  readonly attribute unsigned short width;
  readonly attribute unsigned short height;
  attribute Storage preferences;
  attribute Function onmodechange;
  boolean hasFeature(in DOMString url);
  void openURL(in DOMString url);
  void getAttention();
  void showNotification(in DOMString title, in DOMString message, in Function
onclick);
}
```

8.3.4.2 Wookie extended features

These features are made available by Wookie as extensions to the W3C Widgets specifications, and are not yet supported by other implementations.

8.3.4.2.1 ATTRIBUTES

8.3.4.2.1.1 Widget.instanceId

The read-only identifier generated by the widget engine for this widget instance.

8.3.4.2.2 METHODS

Widget.sharedDataForKey(key, callback)

Returns the value of shared data for *key*, or undefined if there is no match. When completed, invokes *callback* with the return value.

Widget.setSharedDataForKey(key, value, callback)

Sets the value of shared data for *key* to *value*, overriding any existing value. . If there is no match, creates a new shared data entry for *key* with *value*. When completed, invokes *callback* with the return value.

Widget.appendSharedDataForKey(key, value, callback)

Appends the value of shared data for *key* with *value*. If there is no matching key, creates a new shared data entry for *key* and sets it to *value*. When completed, invokes *callback* with the return value.

Widget.proxify(url)

Returns the proxified version of *url*.

Widget.lock()

Sets the state of the widget to locked.

Widget.unlock()

Sets the state of the widget to unlocked.

8.3.4.2.3EVENTS**8.3.4.2.3.1.1 Widget.onSharedDataUpdate**

Called when a shared data entry is updated with the array of shared data keys affected.

8.3.4.2.3.1.2 Widget.onLocked

Called when the state of the widget is set to Locked.

8.3.4.2.3.1.3 Widget.onUnlocked

Called when the state of the widget is set to Unlocked.

8.3.5 Appendix B to Wookie Widget Developers Guide: Server API exposed though the web

The full API of the widget server is pretty simple and has only 5 calls. Each call has to pass a number of arguments as name/value pairs.

8.3.5.1 getwidget

Description

Either creates a new instance of a particular widget or returns a previously used instance.

Parameters

requestid (uses the api call getwidget)

userid (unique for a given user participating within a particular widget instance)

shareddatakey (can be any string)

api_key (A string id value which is also present in the ApiKey table of the widget server)

servicetype (a string value such as 'chat', 'forum' or 'vote') If the widget type does not exist, the server will return a dummy widget "not supported".

Example

```
http://localhost:8080/wookie/WidgetServiceServlet?requestid=getwidget
&userid=paul&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat
```

Returns

An XML fragment with information on how you should display the widget in your calling application.

```
<widgetdata>
<url>

http://localhost:8080/wookie/wservices/www.tencompetence.org/widgets/
default/chat/index.htm?idkey=N9xrPyhLWZUC7UXnWsjVKYbr8bE.eq.&proxy=ht
tp://localhost:8080/wookie/proxy&st=wookie%3AyoGtWfi05DhnluULQ1B3i48C
8iJcSP70A%2F06HUrQ4SJux%2FVnZdM84Yhrz1CZjd6VbckJRGm6yNPWlnqtoOn0SSlY
J6iq15VFAoo%2Bzn37w29xYMCq%2Bdi%2BOEYoV7kHd%2F9ZkFhSR9r514b9o9036Ir4F
MkQ1bht6Gg2wOnXCshIBmjfAzEWCpDWUmoUIOaVF8LLX2C9g%3D%3D

</url>
<title>Default chat widget</title>
<height>358</height>
<width>500</width>
<maximize>false</maximize>
</widgetdata>
```

<url> to where the widget instance can be found

<height> to show the widget instance in your client application

<width> to show the widget instance in your client application

<maximize> to flag if the widget can be maximized to full screen or if it should be fixed to the default height & width in the client application.

Additional information

The shareddatakey parameter is used to form a widget instance. If another user wants access to the same chat instance for example, then for that user you would make the same call as above, but change the userid parameter to the second chat users id. For example...

`http://localhost:8080/wookie/WidgetServiceServlet?requestid=getwidget&userid=phil&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat`

Now both paul and phil have access to the same chat instance.

8.3.5.2 setpublicproperty

Description

Sets a value for a given widget instance, which all other users of the instance have access to. In essence this is shared data between instances.

Parameters

requestid (uses the api call setpublicproperty)

userid (unique for a given user participating within a particular widget instance)

shareddatakey (can be any string)

api_key (A string id value which is also present in the ApiKey table of the widget server)

servicetype (a string value such as 'chat', 'forum' or 'vote') If the widget type does not exist, the server will return a dummy widget "not supported".

propertyname (a value that a particular widget defined to store shared data)

propertyvalue (the value to set it to)

Example

`http://localhost:8080/wookie/WidgetServiceServlet?requestid=setpublicproperty&userid=testuser&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat&propertyname=defaultChatLog&propertyvalue=clearThisPlease`

Returns

An XML fragment with a completed message.

`<message>completed</message>`

Additional information

If the named property does not exist, then the widget server will try to create it. If the property is NULL then the widget server will delete the named entry.

8.3.5.3 setpersonalproperty

Description

Sets a value for a given widget instance, which only the named userid has access to. In essence this is preference data for one user, in one instance.

Parameters

requestid (uses the api call setpersonalproperty)

userid (unique for a given user participating within a particular widget instance)

shareddatakey (can be any string)

api_key (A string id value which is also present in the ApiKey table of the widget server)

servicetype (a string value such as 'chat', 'forum' or 'vote') If the widget type does not exist, the server will return a dummy widget "not supported".

propertyname (a value that a particular widget defined to store shared data)

propertyvalue (the value to set it to)

Example

http://localhost:8080/wookie/WidgetServiceServlet?requestid=setpersonalproperty&userid=testuser&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat&propertyname=moderator&propertyvalue=true

Returns

An XML fragment with a completed message.

`<message>completed</message>`

Additional information

If the named property does not exist, then the widget server will try to create it. If the property is NULL then the widget server will delete the named entry.

8.3.5.4 stopwidget**Description**

Allows a widget instance to be locked, so that no further additions to the shared data may be made. An example of this is when a moderator wishes to lock a chat service, so that no other users can send messages.

Parameters

requestid (uses the api call stopwidget)

userid (unique for a given user participating within a particular widget instance)

shareddatakey (can be any string)

api_key (A string id value which is also present in the ApiKey table of the widget server)

servicetype (a string value such as 'chat', 'forum' or 'vote') If the widget type does not exist, the server will return a dummy widget "not supported".

Example

http://localhost:8080/wookie/WidgetServiceServlet?requestid=stopwidget&userid=testuser&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat

Returns

An XML fragment with a completed message.

```
<message>completed</message>
```

Additional information

It is up to the particular widget itself to handle the “onlock” event.

8.3.5.5 resumewidget**Description**

Allows a widget instance to be unlocked, so that users can make additions to the shared data. An example of this is when a moderator wishes to unlock a chat service, after previously calling “stopwidget”

Parameters

requestid (uses the api call resumewidget)

userid (unique for a given user participating within a particular widget instance)

shareddatakey (can be any string)

api_key (A string id value which is also present in the ApiKey table of the widget server)

servicetype (a string value such as ‘chat’, ‘forum’ or ‘vote’) If the widget type does not exist, the server will return a dummy widget “not supported”.

Example

```
http://localhost:8080/wookie/WidgetServiceServlet?requestid=resumewidget&userid=testuser&shareddatakey=mysharedkey&api_key=TEST&servicetype=chat
```

Returns

An XML fragment with a completed message.

```
<message>completed</message>
```

Additional information

It is up to the particular widget itself to handle the “onunlock” event.

8.4 Appendix 4: Evaluation of the LD Toolkit

8.4.1 Objectives

This evaluation action added to previous evaluations by using the integrated LD Toolkit in a near final version (integrated LD QTI authoring, integrated LD and QTI runtime, publishing UOLs on the the PCM).

Earlier evaluation of ReCourse had provided authors with a short training input, after which they were able to create simple UOLs (see D6.2), and this was born out by a focus group with authors conducted in May 2009. In the present evaluation no such training was provided, in order to provide a more demanding test of usability over a longer period of time (such as may be faced by many users outside TENCompetence). On the other hand the users involved had a stronger technical background than earlier evaluation actions.

Thus the objectives of the evaluation were

- a) to thoroughly test key components of the LD Toolkit in order to identify outstanding functionality and usability issues prior to the final release.
- b) to assess the use of the integrated LD Toolkit in an unsupported environment.

8.4.2 Evaluation methodology

Table 1 indicates the different data sources considered to evaluate the WP6 tools (ReCourse, QTI editor, LinkTool and NewAPIS). A mixed evaluation methodology, combining qualitative and quantitative data gathering techniques, was followed. Quantitative data were considered useful for doing statistics. Qualitative results were used to confirm or reject the problems encountered, to understand them, and to identify future work.

Appendix 1, Table 1: Data sources for the evaluation of the WP6 tools and labels used in the text to quote them

Data source	Type of data	Labels
<i>Post-test questionnaire</i>	Quantitative and qualitative participant characteristics, expectations and evaluation.	[pre-test] [post-test]
<i>GoogleDocs tables</i>	Qualitative participants' problems after interacting with the tools (03.12.09)	[tables-problems]
<i>Interview with participants</i>	Qualitative: participants' opinions.	[interview]
<i>User guide</i>	The guide explains the steps that the users had to follow in order to create, publish and interact with a UoL.	[user-guide]

In order to evaluate the WP6 tools, 2 actions were performed:

Action 1: Users interact with the WP6 tools and indicate the problems encountered

A GoogleDocs document which contained 4 different tables [tables-problems] (see *Annex: Data collection, section 3.1*) where the participants had to add the problems encountered after interacting with the ReCourse authoring tool, the QTI editor, the LinkTool and newAPIS. This document was used to collect quantitative and qualitative data.

Number of participants: 8 participants.

Methodology: the participants have to create a UoL with a QTI test. Then they have to publish it and interact with the UoL and the test.

Time: Participants had two weeks to test the tools.

Support: They had to follow a guide [user-guide] (see *Annex: Data collection instruments, section 3.3*) which explains the steps to create, publish and interact a UoL with a QTI test.

Collection of results: The participants had to complete 4 tables with two columns (problem and description of the problem). In these tables they had to add the problems encountered during the testing of: the QTI editor, ReCourse, LD Runtime and newAPIS.

Action 2: Interview with participants and post-test

A post-test (see *Annex: Data collection instruments, section 3.2*) was used to collect the final comments of the participants. Finally an interview with the participants was performed to understand their profile.

Number of participants: 6 participants.

Methodology: a post test with open and closed questions was delivered to the participants. A short interview was done to understand the profile of the participants.

Time: Participants had 1 hour to answer the questionnaire.

Collection of results: The collected results were compared and analyzed. The results are discussed in the next section.

Analysis

Once the problems were added to the tables [tables-problems] the problems encountered were analyzed by the developers of the tools. A new column “status” was added to the tables, the developers had to indicate the priority for solving the problem encountered.

The description of the different elements of the table was:

Problem: descriptive name of the problem.

Description: Sentence with gives a description of the problem.

Status:

Priority: an indication of the seriousness of the problem

- 2 **major functionality problem** (i.e. authors cannot use the system to achieve their goals)
- 3 **minor functionality problem** (i.e. authors need to use workarounds, or minor aspects are not possible to achieve)
- 4 **major usability problem** (i.e. it is very difficult for target users to use the system, even after it is explained to them)
- 5 **minor usability problem** (i.e. it is not intuitive for target users, but once they know how it works they can use it).

The list of problems encountered is set out in 3. *Data collection instruments section 3.1.*

The post-test contained open and closed questions and it was structured considering the main functionalities of the tools. The main objective was to evaluate the usability aspects of ReCourse, the QTI editor, the LinkTool and newAPIS. 6 of the total of 8 participants answered the post-test.

In the next discussion some of the results will be structured as in the post-test by tool and functionality:

Participant profile

A total of 8 participants test the WP6 tools. 8 participants indicated the problems encountered during the experience of creating, publishing and interacting with a UoL. 6/8 participants answered the post-test.

The profile of the participants was: 5 men and 3 women between 24-29 years interacted with the tools. They have technical skills (all the participants were computer science engineers) [interview]. 2 of them had not had contact with the tools before the start of the evaluation process.

8.4.3 Tabular Results

ReCourse Learning Design Editor	
Functionality	Results
Adding a new Module/Phase	5/6 participants indicated that they did not have problems using this functionality. 1 participant indicated that it is a easy task, but for him is a problem when a module has been created it does not appear as selected.
Adding a new Role	3/6 participants explained that they have problems trying to delete a role created. 2/6 participants commented that the first time they used the tool was difficult to find the button to add a new role. A possible future work would be to add a new button for managing (create, delete) the roles
Adding a new Activity	6/6 indicated that it is an easy task. 1 participant commented that it would be very useful to add a description to distinguish between the different possible activities.
Using the environment section	4/6 participants do not have problems using this section. 2 of them indicated that it is easy and intuitive to use the section called “Environments”. Other 2 participants (with more experience using the tool) explained that the step which is not intuitive is how to associate environments and activities. 1/6 participant (the one with less experience) commented that it is difficult to understand what is the meaning of some of the terms of this section, this fact difficult the management of the elements. S/he suggested to add notifications to help the users. Finally one participant indicated that s/ he does not have experience using this section.
Using the resources section	In general participants had no problems using this section. One participant had problems for distinguish the names of the different resources (resources for this unit of learning, resources dependent files, files...). Two participants indicated that when they delete a resource it does not means that the resource disappears because some times it produces consistency errors.
Managing properties/Conditions	Only one participant had experience using this section. S/he indicated that there are too many options and it is difficult to manage them. S/he suggested adding notifications to help users when they are using properties. Another aspect is to add some function for communicating QTI with LD properties.
Packaging the UoL	No participant had problems packaging the UoL.
Others aspects	Griffiths, D. and O. Liber, Opportunities Achievements and Prospects for use of IMS LD, in Handbook of Research on Learning Designs and Learning Objects, L. Lockyer, et al., Editors. 2008, Information Science Reference. 4/6 participants commented that the vocabulary is aligned with the IMS LD notation, they commented that it would be useful to add some manual or help for beginners. The appreciation of the tool was good. Participants are in general satisfied using ReCourse to create a UoL, they commented that it is very useful for adding widgets. If the user has knowledge about the IMS specification then ReCourse is a very useful tool to create UoLs. They have the possibility of rating the tool (1 (not useful) – 5 (very useful))

	<i>4/6 participants rate the tool with 4 points and 2/6 participants rate the tool with 3 points.</i>
--	---

QTI Editor	
Task	Results
Adding a new Activity Test	<i>No problems detected with this task. One participant suggested adding a separate button to create the QTI tests.</i>
Editing Multiple Choice	<i>No problems detected with this task.</i>
Editing Multiple Response	<i>No problems detected with this task.</i>
Editing a Yes/No question	<i>No problems detected with this task.</i>
Editing Fill in the blank	<i>All the participants had problems editing this question. Creating a blank is not intuitive. Participants need a previous explanation in order to know how they can use this question.</i>
Editing Likert	<i>No problems detected with this task.</i>
Editing Open question	<i>No problems detected with this task. One participant suggested to use this type of question to send messages between students and teachers.</i>
Editing InlineChoice	<i>5/6 participants did not use this question. 1/6 participants indicated that s/he has the same problem using this question as the indicated with the fill in the blank.</i>
Editing Match question	<i>Editing this question is not intuitive. Participants need a previous explanation in order to know how they can use this question.</i>
Editing Gap-Match question	<i>Editing this question is not intuitive. Participants need a previous explanation in order to know how they can use this question.</i>
Editing Order question	<i>No problems detected.</i>
Editing “Associate” question	<i>No problems detected.</i>
Editing Hot-text question	<i>Editing this question is not intuitive. Participants need a previous explanation in order to know how they can use this question.</i>
Editing Slider question	<i>No problems detected. One participant commented that s/he did not understand the difference between a likert question and a slider question.</i>
Editing report results	<i>6/6 Participants indicated does not have problems using this section. Some comments were that when they have to add the score (editing the feedback) if you do not have previous experience the sentence “if the score is larger than” it sentence creates doubts.</i>
Saving the test	<i>No problems detected.</i>

Other aspects	<p>Add some explanations about the use of the score (it is possible to add negative scores?).</p> <p>Participants indicated that they did not know how they can sort the questions. One participant had problems deleting a question. The appreciation of the tool was good. Participants are in general satisfied using the QTI editor to create tests. They really appreciate the possibility of adding feedback to the questions and to the test.</p> <p>They have the possibility of rating the tool (1 (not useful) – 5 (very useful))</p> <p>4/6 participants rate the tool with 3 points and 2/6 participants rate the tool with 4 points.</p>
----------------------	---

NewAPIS player of QTI questionnaires	
Task	Results
Answering a question	<i>No problems detected.</i>
Understanding which were the answered questions	<p><i>No problems detected.</i></p> <p><i>One participant commented that the colour “grey” for the answered questions could be changed for another one more colourful.</i></p>
Using the buttons “Previous” and “Next”	<i>No problems detected.</i>
Understanding the results report	<i>No problems detected.</i>
Other aspects	<p><i>The participants commented that the simplicity of the player allows them to interact with the questionnaires without problems.</i></p> <p><i>The appreciation of the tool was very good.. They have the possibility of rating the tool (1 (not useful) – 5 (very useful)).</i></p> <p><i>5/6 participants rate the tool with 4 points and 1/6 participants rate the tool with 5 points.</i></p>

8.4.3.1 Comments and conclusions

Participating authors found that they could create, publish and interact with the UoLs successfully, as summarised by one participant who said: *“Both ReCourse and the QTI editor really facilitate the creation of UoLs and QTI tests by avoiding the tedious work of dealing directly with the specification”* [post-test].

In general the user interface of ReCourse interface was very well received, and it helped in managing the tasks involved in creating a UOL.

Some issues were identified with functionality and interface, but participants commented that the majority were not critical. The only issue shared by a number of participants was that the development team might consider separating *roles* and *activities* onto different buttons.

All participants commented that in some areas these tools used rather technical language. This is not necessarily a problem given that some functionality is intended for advanced users. However, it seems that it would be very useful to expand explanations about terminology (or change them where appropriate) and provide more guides for beginners [post-test, interview]. One participant suggested adding more templates of UoLs [post-test]³³.

The detailed feedback on user interface and functionality provided input to the preparation of the final release of the Learning Design Toolkit. All the major usability and functionality problems identified, which largely concerned QTI aspects, have been addressed in the final release of the software.

³³ In the version of the LD Toolkit used in this evaluation the full set of TENCompetence templates was not available

8.4.4 Data collection instruments

3.1 Tables with problems, description of the problem and status

Tool: QTI - Editor

Task: Creating a QTI test

Problem	Description	Status
Question Fill in the blank	The creation of blanks in the "fill in the blanks" question type is not intuitive	Priority: major usability problem Week:
Questions information	The "i" (information) button for each question type does not provide any info (image) yet	New images have been created: Associate, order, match, likert Pending: Gap Match No images for: Inline, Hot text and Slider
Question Yes/No	Avoiding to add more choice options in "Yes/No question"	Priority: minor functionality problem
Questions support	Adding more explanations to support the creation of certain question types	Priority: minor usability problem
Question inline choice	The edition of this question is not intuitive	Priority: major usability problem
GapMatch question	In previous versions of the QTI editor the code generated was well interpreted by newAPIS. Now it has changed because newAPIS does not render/support interaction of this question.	Priority: major functionality problem & major usability problem Week:
Score Edition	if the author does not add any score, writes the editor "0" or other score by default?	Priority: minor functionality problem
Delete Question	The image of the litter for deleting is not very clear	Priority: minor usability problem
Modal-Feedback	When doing the modal feedback "if the score is larger than ..." is told but in the output xml , the variable which is going to be compared against the tables is defined as a percentage. <pre><lookupOutcomeValue identifier="SUMMARY"> <product> <baseValue baseType="float">100.0</baseValue> <divide> <numberCorrect /> <numberPresented /> </divide> </product> </lookupOutcomeValue> </outcomeProcessing> <testFeedback access="atEnd" showHide="show" outcomeIdentifier="SUMMARY" identifier="TF-660568eb-52ac-4d63-a8c0-f2119cc71d3b-168" title="Summary Feedback"> <p>Very good results!</p></pre>	Priority: major functionality problem
Score	Is it possible to add score like 0.5 or 42 in multiple choice questions?	Priority: minor functionality problem

Tool: ReCourse**Task: Creating a UoL with QTI test**

Problem	Description	Status
Level B	When a test is created in a UoL, it would be helpful to change automatically the level of the UoL.	<i>This is already implemented - when a new Test Activity is added, and the UOL is saved, you will notice that the UOL is automatically changed to Level B if it was previously Level A</i>
ReCourse- wizard	The wizard functionality is very valuable. Very few people will start creating a UoL from scratch	
New test activity	If you only have one role created, and this role is "Teacher", the checkbox of "role that perform this activity" appears disabled and no selected. Can not continue without choosing a second role.	<i>This is fixed for the next version of ReCourse, but please note that a UOL must have at least one Learner Role</i>
Zip- Package1	Maybe the "package to zip file" deserves its own button on the taskbar	<i>Consider whether this is useful, it may be not so useful if the main work-flow is that the user uploads the UOL to a server.</i>
Zip-Package2	"Include additional ReCourse Information" checkbox'. What does it means?	<i>I will add an explanatory text.</i>

Tool: LD Runtime**Task: Publishing a UoL with a test**

Problem	Description	Status
Vocabulary	Very technical vocabulary: run, UoL, ...	Priority: major usability problem
EXCEPTION	Has the same orange color that WARNING	Priority: minor usability problem
Java_home	Missing information about the JAVA_HOME . Problems with the java home....	Priority: major usability problem
Linktool-administrators access	Accessing to the publish functionality by clicking on a dot (".").	Priority: major usability problem

Tool: NewAPIS

Task: Render and Interact with a UoL with a test

Problem	Description	Status
Questions_answered	Red responded answers in Explorer / Grey in Mozilla	Priority: minor usability problem -
Multiple_cardinality	Fill in the Blank and Gap Match do not work with multiple options.	Priority: minor functionality problem
Score_formula	Change the formula of computing the final score and percentage of corrects Editor: yongwu has to solve the xml mistakes detected by Lau newAPIS: waiting Yongwu's solution	Priority: Major functionality problem
Formula report results	The formula to calculate the interpolation tables is not used from the xml. Score is used instead (depending of problem score_formula)	Priority: Major functionality problem
Next button	Locate Next button at the bottom-right part of the questionnaire (instead off the bottom-left part)	Priority: minor usability problem
Language_feedback	The English text of the outcome is not readed from the xml Language policies have to be discussed	Priority: Minor functionality problem
Questions supported	Multiple choice --> render: ok // interact: ok likert--> render: ok // interact: ok open question--> render: ok // interact: no Match--> render: ok // interact: no Yes/no--> render: ok // interact: ok Order--> render: ok // interact: no Fill in the blank--> render: ok // interact: ok Multiple response--> render: ok // interact: ok associate--> render: ok // interact: no Gap Match--> render: no // interact: no Hot text--> render: no// interact: no Slider--> render: no // interact: no Inline Choice--> render:no //interact:no Gap Match: in previous versions of the QTI editor this questions Works well,not now. Trying to do that questions which newAPIS only renderize also enables the interac2.12tion	Priority: Minor functionality problem
Properties	Integration of level B properties (score).	Priority: Major functionality problem

3.2 Post-questionnaire

Evaluation of ReCourse, LD Runtime and NewAPIS

This is a questionnaire for evaluating the usability aspects of ReCourse, LD Runtime and NewAPIS. If you have any comment (positive or negative) you can add it in the “Comments” column or at the end of the questionnaire.

1. Personal Data

1.1. Gender:

Man / Woman

1.2. Age:

20-25 / 25-30 / 30-35 / 35-40 / 40-45 / 45-50 / 50-55

1.3. Did you have previous experience with the following tools?

ReCourse: Yes / No

LinkTool: Yes / No

newAPIS player: Yes / No

2. Interacting with ReCourse

2.1. Problems creating a UOL

Task	Yes	No	Comments: <i>Add here some problem or comment</i>
Adding a new Module/Phase			
Adding a new Role			
Adding a new Activity			
Using the environment section			
Using the resources section			
<i>Managing properties/Conditions</i>			
<i>Packaging the UoL</i>			

<i>Others</i> <i>(example: vocabulary, other functionalities...)</i>			
--	--	--	--

2.2. Problems creating a QTI test

Task	Yes	No	Comments: <i>Add here some problem or comment</i>
Adding a new Activity Test			
Editing a Multiple Choice question			
Editing a Multiple Response question			
Editing a Yes/No question			
Editing a Fill in the blank question			
Editing a Likert question			
Editing a Open question			
Editing a InlineChoice question			
Editing a Match question			
Editing a Gap-Match question			
Editing a Order question			
Editing a Associate question			
Editing a Hot-text question			
Editing a Slider question			
Editing the report results			

Saving the test			
Others (example: vocabulary, other functionalities...)			

2.3. Rate ReCourse: 1 / 2 / 3 / 4 / 5

3. Interacting with the LinkTool

3.1. Please indicate if you had any problem with the following tasks

Task	Yes	No	Comments: <i>Add here some problem or comment</i>
Publishing a UoL			
Messages (warning, error, correct)			
Creating a Run			
Adding users to the Run			
Join a Run			
Others (example: vocabulary, other functionalities...)			

3.2. Rate the LinkTool: 1 / 2 / 3 / 4 / 5

4. Interacting with the player of QTI tests

4.1. Please indicate if you had any problem with the following tasks

Task	Yes	No	Comments: <i>Add here some problem or comment</i>
Answering a question			
Understanding which were the			

answered questions			
Using the buttons “Previous” and “Next”			
Understanding the results report			
<i>Others</i> (example: vocabulary, other functionalities...)			

4.2. Rate the player: 1 / 2 / 3 / 4 / 5

Thank you for your collaboration!

3.3 User-guide: Creating, publishing and interacting with a UoL

Testing ReCourse, LD runtime toolkit (newAPIS, Coopercore) and LinkTool

Info:

The LD runtime toolkit (specifically the QTI engine newAPIS) only interprets the following types of questions: Multiple Choice, Multiple Response, Yes/No, Fill in the Blank (only with one blank), Likert.

The editor offer more types of questions but if you want to create questionnaires with these questions, you can not use the LD runtime to interpret these questionnaires.

Please when you edit the Report Results of the QTI editor, you have to select all the check-boxes.

Phase 1: Creating a UoL with a QTI test

- Download the authoring tool ReCourse. Link: <http://www.tencompetence.org/ldauthor/>
- Unzip the folder. Press over the icon called “ReCourse”.
- Go to “File” and select “New” and “Learning Design”.
- A new Windows with three options appear. Select “Empty”.
- Introduce a name (e.g. UoL_test) and press the button “Finish”. And Start using recourse.
- In this Windows you have to introduce a name of the UoL. (e.g. “My first UoL”) and be careful if you want to add a QTI test in this UoL, you have to select in the “Level” menu, the option “B”.
- At the bottom of this window, you can see 4 tabs. “Overview, Design, Environments and Resources”.
- After completing the “Overview” page, now you can press over the “Design” tab.

- A new window appears. You have to press over the “New” button, and select “Module”. Then a new module appears in the modules column.
- Next step is to add a role and an activity. You have to press over the other “New” button. Select the “Learner” role. Then you have to introduce the name of this role (you don’t have to full fill the other parameters).
- Now a new “Role” row appears. The next step is to create a “Test Activity”. You have to press in “New” and select “test activity”. Introduce a name and press “Finish”.
- The QTI-editor appears. Now you can create you own questionnaire.
- You have to select a type of question (“Question type” menu). Then press over the button “Add question”. Different parameters will appear depend on the question selected. For each question: you have to select the correct question, add a score for each option, and a feedback message.
- To add a new question. You have to select a new (or same) type of question, and press “Add question”. You can repeat the steps (13 and 14) ever you wanted to add new questions.
- When you have added your last question, then you have to press over the button “Report Results” (you can see this option at the top of the window).
- A new window appears. You have to select the options that you want to show in the report result, and press the “Add feedback” button if you want to show feedback which is related with the final result obtained by the user.
- When you have finished the test. You can close the tab of the editor.
- Then you can see the “Design” window. To finish you have to press over the icon of the “floppy disc” and save your changes. Or you have to go to “File” and “Save”.
- The final step is to “Package to zip file” (you can see this option in “File”), you have to browse where you want to save the package and introduce a name.

Now you have created a UoL with a QTI test! You have to publish it and interact with the UoL!

Phase 2: Publishing and interacting with a UoL with a QTI test

- Download the LD runtime toolkit. Link:
http://www.tencompetence.org/ldruntime/downloads/ccrt_3.5_RC1_tenc_update_v1_3_3_hsql.zip
- Unzip the folder. Press over the icon “start_coppercore”. A console appears, you have to wait some seconds.
- Open a browser and put: <http://localhost:8080/linktool/> . The linktool appears. To Publish a UoL you have to press the “.” That you can see after “2008”.
- A new window appears. You have to introduce “sledpass” as password.
- Select the “publish new” option. A new window appears. You have to browse the UoL that you have created with ReCourse. And press “publish”. Don’t worry if some orange warnings appear.
- Now you have to press the “UoL” tab. And you can see your UoL.
- Now you have to test your UoL. Press over the “logout”(top of the page).
- You can see the main page of the LinkTool. Now press over the “register” option. And fill in the different parameters (you can introduce a fake email address).
- Now you can introduce your user-name and password.
- You can see your UoL. Select it.

- Introduce a name of the run. (The start time is not necessary). Press “Create Run”.
- Press over the name of your run.
- A new window appears. Press over the sentence “Click here to join”. Then press over “Go to player”.
- Finally you can interact with your UoL. Press over the Test activity. And the different questions will appear. You can answer the questions and see the feedback.

You have published successfully your UoL and now other users can interact with it!